Raspberry Pi Foundation

Experience AI

AI Glossary of terms

Experience AI — Glossary of terms

This glossary explains artificial intelligence (AI) and machine learning (ML) key terms used in the Experience AI Lessons and beyond.

Jump to the glossary

We have designed these explanations primarily for teachers and educators, with a young audience in mind. With this glossary, we aim to support you to strengthen your understanding of these key terms, as well as your technical knowledge.

Vocabulary is an important part of teaching and learning. Using vocabulary correctly can support learners to develop their understanding, while its inconsistent use can lead to alternate conceptions (misconceptions) that can interfere with students' learning. You can read more about this in our Pedagogy Quick Read on alternate conceptions. As a teacher, using accurate, technical vocabulary regularly and consistently can support students' conceptual understanding.

We have used 'semantic waves' theory to help us write the explanations. Each explanation follows the same three-part structure: the first part is a more abstract explanation of the term, the second part unpacks the meaning of the term using a common example, and the third part repacks what was explained in the example in more abstract terms again to reconnect with the vocabulary. You can find out more in our Pedagogy Quick Read on semantic waves.

This is Version 1 of the glossary. This glossary will be added to, revised, and updated alongside the evolution of the Experience AI Lessons.

Table of contents

ML accuracy

Accuracy refers to how correct something is. In **machine learning (ML)**, accuracy is a way of measuring how often an ML **model** is making a correct **prediction**. For example, a **classification** model is designed to classify apples. From 100 images of apples, 90 are classified correctly. The classification accuracy of the model is 90%. Accuracy is one way to evaluate ML models. Most often, accuracy is used along with other measures to evaluate the quality of a model.

Artificial intelligence

Artificial intelligence (AI) is the design and study of systems that appear to mimic intelligent behaviour. Some AI applications are based on rules. More often now, AI applications are built using **machine learning** that is said to 'learn' from examples in the form of **data**. For example, some AI applications are built to answer questions or help diagnose illnesses. Other AI applications could be built for harmful purposes, such as spreading fake news. AI applications do not think. AI applications are built to carry out tasks in a way that appears to be intelligent.

AI literacy

AI literacy is a set of competencies and ways of thinking that allow people to meaningfully engage with **artificial intelligence (AI)** applications, as well as in situations where AI applications are used around them. Such competencies include understanding AI, being involved in developing AI systems, and having informed opinions on the ways in which AI systems are used in the world. For example, a person uses AI literacy when they assess the accuracy of the information provided by an AI chatbot application. As well as different ways of engaging with AI applications, AI literacy includes people's potential to actively participate in deciding how AI systems might be used around them

AI project lifecycle

The **artificial intelligence (AI)** project lifecycle refers to the different steps it might take to design and build a **machine learning (ML) model**. The steps include defining the problem, preparing the **data**, **training** the model, testing the model, evaluating the model, and explaining the model. For example, an ML model is designed to generate new song playlists. An ML developer might first consider what type of playlist they want to create. Next, they might collect and prepare song data. The ML model is trained and tested with the song data. The ML model is evaluated to see if it works as expected. Finally, the ML model is explained so that others can use it. Usually, the AI project lifecycle steps are used iteratively rather than one after the other. The AI project lifecycle is a series of iterative steps used to build and improve an ML model.

Bias

Bias refers to a preference for or against something. For example, a student might prefer English lessons to maths lessons and spend more time on their English homework. In other words, they have a bias towards the subject of English. There are many types of bias, including **societal bias** and **data bias**. **Machine learning (ML)** developers have to think carefully about whether the **data** they are using to **train** ML **models** is biased or not. Being biased can result in giving an advantage to a person, group, or set of ideas or beliefs compared with another.

ML class

To **train classification models**, **machine learning (ML)** developers organise the **data** into predefined groups called classes. Classes are defined in advance based on what people find useful to group things into. Imagine an ML application designed to identify fruits in a supermarket. The data might be organised in classes of apples, bananas, oranges, blueberries, etc. A class is a group of things that classification models use to identify similarities in the data.

ML classification

Classification refers to the task of assigning things into predefined groups, called **classes**. Classes are defined in advance based on what people find useful to group things into. An example of a classification problem is sentiment analysis of song reviews. A **machine learning (ML)** classification **model** is **trained** with reviews **labelled** by people as ‘positive’ or ‘negative’. After training, the ML model can be used to **predict** whether a new review should be classified as either ‘positive’ or ‘negative’. A classification model predicts one or more class labels. A classification approach is useful for solving problems where the answer falls into predefined groups.

Computer vision

Computer vision is the study of systems designed to process information from digital images or videos. Computer vision application examples include facial recognition, medical imaging, and video surveillance. For example, computer vision is used in the design of self-driving cars to detect and avoid crashing into objects. Most often, computer vision systems use **machine learning models** to identify patterns in the image and video data. Computer vision systems are useful where information from digital images or videos can be used to solve a problem.

ML confidence

Confidence refers to how certain something is. In **machine learning (ML)**, confidence is a way of measuring the certainty of a **prediction**. For example, a **classification model** is designed to predict if it will rain tomorrow. The model predicts with 90% confidence that there will be rain tomorrow. In other words, there is a 90% certainty that tomorrow it will rain. Using confidence to measure the certainty of predictions helps to evaluate the quality of an ML model.

ML confidence threshold

Confidence threshold refers to a value set as a level of acceptance for a **machine learning (ML) model's predictions**. The confidence threshold is chosen by the ML developer when designing the ML model. For example, a prediction is generated by an ML model with 50% **confidence** that there will be a snowstorm tomorrow. However, if the confidence threshold is set to 60%, that prediction will be considered inaccurate. In other words, unless the prediction's confidence is 60% or higher, it will not be accepted as **accurate**. The value is set according to the nature of the problem being solved, with medical diagnosis predictions requiring a higher confidence threshold than song recommendations. The choice of threshold value determines what is an acceptable level of prediction confidence.

Data

Data refers to values, measurements, facts, or observations in a form suitable to be processed by computer programs. There are many types of data, such as text, image, or sound. An example of text data is messages people exchange with their friends on digital devices. In **machine learning (ML)**, data represents the examples ML **models** are **trained** with. Collecting, **cleaning**, and structuring vast amounts of data are an essential part of designing ML models.

Data-driven

Data-driven is a way of designing systems using **data** instead of step-by-step instructions. For example, knowing what causes certain diseases is hard, but there is lots of example data. Therefore, designers use the medical data of people affected by the disease to diagnose it. Data-driven systems stand in contrast with **rule-based** systems. Data-driven systems are suitable for solving problems where rules that cover every situation are difficult to produce. Instead, enough examples can be collected to inform a solution.

Data bias

Data bias refers to **bias** reflected in the **data** used to **train machine learning (ML) models**. Data bias can lead to ML models being trained to generate biased **predictions**. For example, some facial recognition models are biased against faces of certain skin tones, because the ML models have been trained using mostly images of faces of one skin tone. There are several potential sources of data bias. These include incomplete data, and data that reflects **societal bias**. Detecting data bias is important to avoid ML models generating biased predictions.

Data cleaning

Data cleaning is a step in preparing the **data** used to **train** a **machine learning (ML) model**. Data cleaning involves identifying and correcting errors in the data. For example, fixing typing errors or removing duplicates in text data are two simple data cleaning tasks. Most often, data is messy and requires more complex cleaning before being used to train ML models. There are many ways of cleaning data depending on the problem and data type. Using clean data is essential for building **accurate** ML models.

ML decision tree

A **machine learning (ML)** decision tree is one type of ML **model**. ML developers use decision trees to structure a set of conditions based on which a **prediction** can be made. The conditions are derived from **features** in the **data**. For example, a decision tree might be used to build a movie recommendation system. The decision tree model is **trained** using many people's movie preferences. During training, conditions are generated based on features such as the movie type, length, or lead actor. The ML model generates a prediction of which movie someone might like to watch next, based on how their preferences follow the conditions in the model. The structure of ML decision trees is generated based on vast amounts of data, and may change if retrained with different data.

ML decision tree node

A **machine learning (ML) decision tree** is made up of nodes. The nodes are linked to form a structure based on which a **prediction** can be generated. There are two types of nodes: decision nodes and leaf nodes. For example, consider a decision tree built to predict the types of stars in our solar system. The decision nodes represent **data features** such as the temperature, radius, colour, or brightness of stars. The leaf nodes represent the star types in the form of prediction **labels**, such as 'Red Dwarf', 'White Dwarf', or 'Brown Dwarf'. Decision tree nodes form the structure required for an ML **model** to generate a prediction.

ML explainability

Explainability refers to the extent to which something can be understood. In **machine learning (ML)**, explainability helps people understand how a **prediction** was produced. For example, ML **decision tree models** are explainable because the **nodes** can be analysed in a way that people can understand. Most ML models are not fully explainable, and some are more explainable than others. Increasing the explainability of a model can help fix problems and fight **bias**.

ML feature

In **machine learning (ML)**, features represent characteristics associated with the **data**. For example, a music data set might have features such as tempo, pitch, energy, or genre. Some ML **models** are **trained** using features to find similarities in the data. Others **predict** new features in the data that people cannot easily observe. Choosing what features to use when training an ML model can make a difference to how well the model works.

Generative AI

Generative AI is a type of **artificial intelligence (AI)** designed to generate content, such as text, images, or sound. There are lots of applications that use generative AI, including the production of art or music, or generating text for chatbots. For example, generative AI art applications can generate an image based on a prompt, such as “make me a picture of a dragon reading a book”. Generative AI art is created using **machine learning models trained** on millions of images of existing art. The resulting images may replicate the style of an artist, without the original artist knowing or approving. Generative AI applications are becoming more and more commonplace and often you cannot tell that generative AI is being used.

ML label

In **supervised learning**, a **machine learning (ML) model** is **trained** using labelled **data**. Each piece of data is annotated with one or more labels that provide information about that data. For example, an ML model is designed to identify bird sounds. Each sound is labelled with the name of the bird that made the sound. The ML model is trained with the labelled sounds and can **predict** the label (bird name) of new sounds. Data is most often labelled by people to provide accurate examples to train ML models with.

Machine learning

Machine learning (ML) is an approach used to design and build **artificial intelligence (AI)** systems. ML is said to ‘learn’ by using examples in the form of **data**, instead of executing step-by-step instructions. In other words, ML applications are **data-driven**. For example, an ML application is used to recognise speech. It is based on many examples of people speaking in different accents and tones of voice. Other ML applications include identifying objects in images or playing complex games. Each ML application is designed to solve a specific problem.

ML model

A **machine learning (ML)** model is used by an ML application to complete a task or solve a problem. The ML model is a representation of the problem being solved. ML developers use vast amounts of **data** representative of a specific problem to **train** a model to detect patterns. The result of the training is a model, which is used to make **predictions** about new data in the same context. For example, self-driving cars are built using ML models to predict when to stop. The models are trained using millions of examples of situations in which cars need to stop. There are many different types of models, using different kinds of data, and different ways of training the models. All ML models are trained to detect patterns in the **training data** to make predictions about new data.

ML model card

A **machine learning (ML)** model card is a way of documenting essential information about ML **models** in a structured way. ML model cards are written by ML developers for both experts and non-experts. For example, an ML application is developed to translate different languages, such as from Arabic to French and vice versa. A model card includes information on the model's translation **accuracy**, as well as the model's performance around jargon, slang, and dialects. Other model card information might include the type of ML model, different performance indicators, and even known **bias**. Model cards are created during the explanation stage of the **AI project lifecycle** to expose information on the model's capabilities and limitations, in a way that is easy to understand.

ML prediction

Machine learning (ML) models are **trained** to make predictions. The prediction produced by an ML model suggests what the **data** represents, or what might be useful for a task. For example, an ML developer might train a model to predict which movie someone might want to watch next, based on their viewing habits. The model will generate a prediction after being trained on the movie choices of lots of people. The main job of an ML model is to make predictions. All ML models make predictions, even if in some cases these predictions are not obvious to the user.

Reinforcement learning

Reinforcement learning is one approach used to **train machine learning (ML) models**. This approach is used to solve problems with a clear goal, where rewards and penalties are used to reach that goal. Reinforcement learning approaches are used in the design of self-driving cars or to play complicated games. For example, a reinforcement learning model might be used to design an application to play chess. The model is trained to **predict** the moves that maximise the rewards and minimise the penalties towards winning. Reinforcement learning approaches use rewards and penalties to identify strategies of reaching a set goal.

Rule-based

Rule-based is a way of designing systems using a set of predefined rules. For example, a noughts and crosses (tic-tac-toe) program is designed using rules of what moves to make in order to try to win the game. The rules are defined by humans who are usually experts in the domain of the problem being solved. **Artificial intelligence (AI)** systems built using a rule-based approach are also known as 'good old-fashioned AI'. Rule-based systems stand in contrast with **data-driven** systems where **data** is used as examples of how to solve the problem. Rule-based systems are useful for solving problems where rules that cover most situations can be produced and followed.

ML test data

In **machine learning (ML)**, test data refers to the **data** used to test and evaluate **trained** ML **models**. For example, an ML model is trained to **predict** a diagnosis of a medical condition. Before being used in real-life situations, the model is tested and evaluated using test data. The test data is separate from the **training data** an ML model is trained with. Test data is used to measure the performance of an ML model with examples beyond the training data.

ML training

Machine learning (ML) models are trained using examples in the form of **data** to find patterns and make **predictions**. During training, the patterns are fine-tuned to improve predictions. For example, an ML developer might build a model to recommend songs. The ML model will be trained on many people's song choices to find similarities between what different people enjoy listening to. The more diverse song choices the model is trained with, the better the recommended song prediction is likely to be. There are many different ways of training ML models, using different types of data. A developer will choose between the types of training available depending on the problem they are trying to solve and the data available to solve it. The quality of training largely depends on the quality of the data used.

ML training data

In **machine learning (ML)**, training data refers to the examples in the form of **data** used to **train** ML **models**. ML developers build models to work out patterns in the training data, which can be used to generate **predictions** about new data. For example, an ML developer builds a speech recognition application. The training data might include lots of examples of people speaking, in different accents or tones of voice. The more the training data represents reality, the better the model is likely to perform.

Societal bias

Societal bias refers to **bias** held by a large group of people, or by society at large. There are many different types of societal biases, such as racial bias, gender bias, or ethnic bias. An example of gender bias is the idea that women are less well suited to engineering careers than men. **Data** collected from large groups of people could reflect societal biases, resulting in **data bias**. If data that reflects societal bias is used to **train machine learning (ML) models**, this may lead to the models generating biased **predictions**. In ML, it is important to mitigate societal biases reflected in the **training data**, to avoid discriminatory or unjust outcomes.

Supervised learning

Supervised learning is one approach used to **train machine learning (ML) models**. Supervised learning approaches use large amounts of **data labelled** by people with relevant information. One type of supervised learning is **classification**. An example of a classification problem is identifying tigers in the wild. The data consists of many images, with the ones containing tigers labelled as such.
The ML model is trained with the labelled images and **predicts** if there is a tiger in those images. Having correctly labelled images allows the developer to know to what extent the predictions of the model are **accurate** and adapt the training of the model. Following that, the ML model can be used to predict if there is a tiger in completely new images. Supervised learning approaches depend on having enough correctly labelled data to produce accurate predictions.

Unsupervised learning

Unsupervised learning is one approach used to **train machine learning (ML) models**. ML developers train unsupervised learning models to organise **data** based on similarities. This process results in finding hidden patterns in the data. One type of unsupervised learning is clustering. An example of a clustering problem is **predicting** how health data can be grouped to help diagnose diseases. These groups are called clusters, which are not known in advance. The ML model can be used to predict if new health data falls into one of the clusters. Unsupervised learning approaches can be useful in solving problems where people may not know what to look for.

Experience AI

This glossary has been created by the Raspberry Pi Foundation as part of our educational programme, Experience AI. Find out more at **www.experience-ai.org**

Raspberry Pi
Foundation