



TASK

Machine Learning

Visit our website

Introduction

Imagine teaching a child a new skill. You might show them examples, provide guidance, and correct mistakes. Machine learning is similar. We teach computers to learn from data, just like a child learns from experience.

In this lesson, we will explore different ways to teach computers: supervised, unsupervised, and semi-supervised learning. We will also discuss how to measure how well a computer has learned a task.



A note from the
HyperionDev Team

The history of machine learning

For decades, visions of machines that can learn the way humans can have captured the imagination of science-fiction authors and researchers alike. However, only in recent years have machine learning programs been developed that can be applied on a wide scale and influence our daily lives.

Machine learning programs are working behind the scenes to produce and curate our playlists, news feeds, weather reports, and email inboxes. They help us find restaurants, translate documents, and even meet potential dates. From a business perspective, machine learning-based software is becoming central to many industries, generating demand for experts.

In 1959, Arthur Samuel, a pioneer in artificial intelligence and gaming, defined machine learning as the "field of study that gives computers the ability to learn without being explicitly programmed." He is known for developing a program capable of playing checkers. Samuel never programmed exactly which strategies the systems could use. Instead, he devised a way in which the program could learn such strategies through the experience of playing thousands of games.

In the 50s, machines were hard to acquire and not very powerful, so machine learning algorithms were mostly an object of theoretical research. Now that computers are vastly more powerful and more affordable, machine learning has become a very active field of study with a variety of real-world applications.



DEFINING MACHINE LEARNING

At its essence, machine learning (ML) can be defined as a computational methodology focused on deriving insights from data. It enables computers to acquire knowledge from past observations and independently make predictions or decisions without relying on explicit programming instructions. By leveraging data-driven patterns and algorithms, machine learning enables automated systems to adapt and improve their performance over time.

The term “machine learning” is often used interchangeably with the term “artificial intelligence” (AI). While the two are very much related, they are not the same thing. There is much debate about the difference between the two, but a simple way to look at it for our purposes is to see Machine Learning as a type of artificial intelligence. Any program that completes a task in a way that can be considered human-like is an example of artificial intelligence, but only programs that solve the task by learning without pre-programming are machine learning programs.

INPUT AND OUTPUT

Whatever it is that we want a machine learning algorithm to learn, we first need to express it numerically. The machine-readable version of a task consists of an **input** and an **output**. The input is whatever we want the algorithm to learn from, and the output is the outcome we want the algorithm to be able to produce. An example of an input would be the budget or number of awards a movie receives. An example of output would be the box office sales of that movie.

Since machine learning is a field that overlaps with several other disciplines, including statistics, the input and output may be referred to by several other names.

For input, these include:

- features (named after the fact that inputs typically ‘describe’ something),
- independent variables, and
- explanatory variables (because the output is usually assumed to depend on or be explained by the input).

For output, alternate terms are:

- labels,
- predictions,
- dependent variables, and
- response variables.

Once we clearly understand the input-output specifics of machine learning models, we can explore different learning algorithms. While there are various types of learning methodologies, our main focus in this context will be on supervised and unsupervised learning. These two approaches are fundamental in the field of machine learning, as they involve training models using labelled or unlabelled data, respectively.

SUPERVISED LEARNING

In supervised learning problems, a program predicts an output given an input by learning from pairs of inputs and outputs (labels); that is, the program learns from examples that have had the right answers assigned to them beforehand. These assignments are often called **annotations**. Because they are considered the correct answers, they are also called **gold labels**, **gold data**, or the **gold standard**.

The collection of data examples used in supervised learning is called a **training set**. A collection of examples used to assess a program's performance is called a **test set**. Similar to a language learner who acquires vocabulary solely through exposure, supervised learning models are trained on a dataset of questions and their corresponding correct answers. Then they must learn to provide the correct answers to new but similar questions.

Next, we will delve into two fundamental supervised learning techniques: regression and classification. Regression predicts continuous values, while classification categorises data into distinct classes.



Take note:

Continuous variables represent data that can take on any value within a given range, like height, weight, or temperature. **Categorical** variables represent data that can be sorted into distinct groups or categories, such as gender, colour, or occupation.

Regression

Regression is a prediction task where a program learns to estimate and predict a continuous output value. It does this by analysing pairs of input features and their corresponding outputs in a training set. By analysing the training examples, the program tries to identify patterns and associations that allow it to make precise estimations.

The main objective of regression is to understand the relationship between the input variables and a continuous target variable, enabling the program to make accurate predictions for new inputs that are similar to the training data.

Classification

Unlike regression tasks, a classification process assumes a program is trained to categorise input data into predefined classes or categories.

By analysing labelled examples, where each example is already assigned to a specific class, the program learns patterns and relationships between input features and classes. This knowledge allows the program to accurately classify new, unseen data, ensuring they are correctly assigned to their respective classes. The ultimate goal of classification is to develop a model that can make reliable predictions for unknown instances, effectively categorising them into the appropriate classes.

To conclude, we offer a list of common supervised learning algorithms and their typical usage:

Supervised learning algorithms	Typical usage	
	Regression	Classification
Linear regression	✓	
Logistic regression		✓
Decision tree	✓	✓
Random forest	✓	✓
Support vector machines (SVM)	✓	✓
Naïve Bayes	✓	
K-nearest-neighbour (KNN)	✓	✓

EVALUATION METRICS

It is important to know how well our model works. Evaluation metrics help us measure a model's performance and compare it to other models.

Metrics for regression models

In regression models, the goal is to minimise the distance between our model's predicted values (denoted as \hat{y}) and the actual observed values (denoted as y).

1. **Mean absolute error (MAE)** is the average absolute difference between actual and predicted values. The closer it is to 0, the better the model's performance, as it indicates a smaller average error between predicted and actual values. MAE is ideal to use when you want to measure the magnitude of prediction errors without emphasising large errors. This makes MAE less sensitive to outliers and ideal for cases where you want all errors to have equal weighting.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Where n is the number of data points and i is the index of a data point.

2. **Mean squared error (MSE)** is the average of the squared differences between actual and predicted values. An MSE close to 0 indicates that the average squared error in the predictions is small. MSE penalises larger errors more heavily than smaller errors due to the squaring of differences, making it sensitive to outliers. MSE is particularly effective when you want to emphasise and penalise large deviations between predicted and actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

3. **Root mean squared error (RMSE)** is the square root of the average of the squared differences between actual and predicted values (MSE). An RMSE closer to 0 signifies a smaller average error between predicted and actual values leading to better predictions. Unlike MSE, RMSE is expressed in the same units as the target variable which can aid in interpretability.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

4. The **R² score**, or the Coefficient of Determination, represents the proportion of variance in the target variable that is explained by the model. This metric helps us assess how well the model captures the variation in the target variable – essentially, how good of a fit the model is. The closer the R² score is to 1, the better the model fits the data.

Metrics for classification models

1. **A confusion matrix** is a table layout used to visualise the performance of classification models. It gives insight into where the model is getting things right, as well as the types of errors it is making when classifying observations. The components of a confusion matrix are frequencies that represent the number of observations classified as:
 - True positives (TP) – model correctly predicted the positive class.
 - True negatives (TN) – model correctly predicted the negative class.
 - False positives (FP) – model incorrectly predicted the positive class.
 - False negatives (FN) – model incorrectly predicted the negative class.

		Predicted class of observation	
		Negative	Positive
Actual class of observation	Negative	TN	FP
	Positive	FN	TP

A confusion matrix is especially useful for binary classification problems and can be extended to multi-class classification problems.

2. **Accuracy** is the ratio of correct predictions to total predictions. The closer it is to 1, the better the model is correctly predicting outcomes across all classes. Accuracy is a good choice when the classes in the dataset are balanced, meaning each class has a roughly equal number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3. **Precision** is the ratio of correctly predicted positives to the total of predicted positives. A precision closer to 1 indicates that most of the positive predictions made by the model were indeed correct. This is useful in situations where incorrectly predicting a positive case (a false positive) may be costly e.g., predicting a non-spam email as spam which can cause inconvenience.

$$Precision = \frac{TP}{TP + FP}$$

4. **Recall** is the ratio of correctly predicted positives to the total of actual positives. A recall closer to 1 indicates that the model is very good at catching all positive cases. This is useful in situations where missing a positive case (a false negative) is costly e.g., predicting that a patient does not have a disease when they actually do.

$$Recall = \frac{TP}{TP + FN}$$

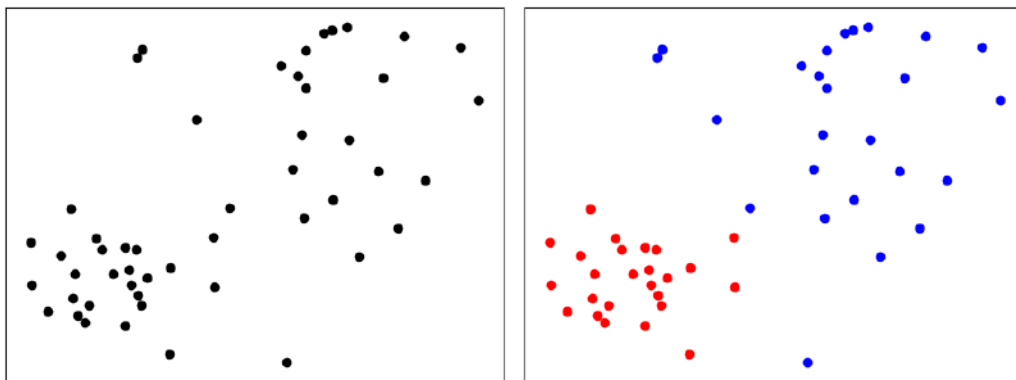
5. **F1-Score** is the harmonic mean of precision and recall. It balances the trade-off between the two and is particularly useful in cases where there is an uneven class distribution.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

UNSUPERVISED LEARNING

In unsupervised learning, a program does not learn from labelled data. Instead, it attempts to discover patterns in the data on its own.

For example, suppose you have two classes scattered in a two-dimensional space (as in the first of the images below) and you want to separate the two datasets (as in the second image on the right-hand side). Unsupervised learning finds underlying patterns in the data, allowing the classes to be separated.



To highlight the difference between supervised and unsupervised learning, consider the following example. Assume that you have collected data describing the heights and weights of people. An unsupervised clustering algorithm might produce groups that correspond to men and women, or children and adults. In supervised learning, some of the data is already labelled. For example, you might tell the algorithm that certain data points are male or female. The algorithm then learns from these labels and can use this information to predict whether a new person is male or female based on their height and weight.

The following algorithms have proven to be highly valuable in practical applications, making them some of the most commonly used methods in unsupervised learning:

- K-means clustering
- Hierarchical clustering
- t-Distributed Stochastic Neighbour Embedding (t-SNE)
- Gaussian Mixture Models (GMM)
- Autoencoders

SEMI-SUPERVISED LEARNING

Semi-supervised learning is an approach that combines labelled and unlabelled data to harness the benefits of both supervised and unsupervised learning. While supervised learning relies on labelled data with known outcomes and unsupervised learning explores unlabelled data to identify patterns, semi-supervised learning uses a smaller set of labelled data to guide the learning process. Simultaneously, it utilises a larger set of unlabelled data containing valuable information.

By leveraging the combined dataset, the algorithm learns from the labelled examples and applies that knowledge to predict outcomes for the unlabelled data, revealing additional patterns and enhancing the model's understanding of the problem's underlying structure. This approach is particularly advantageous when acquiring labelled data is expensive or time consuming, allowing for optimal resource utilisation and the potential for improved results compared to using just one data type.



Take note

The task(s) below is/are **auto-graded**. An auto-graded task still counts towards your progression and graduation. Give it your best attempt and submit it when you are ready.

When you select “Request Review”, the task is automatically complete, you do not need to wait for it to be reviewed by a mentor.

You will then receive an email with a link to a model answer, as well as an overview of the approach taken to reach this answer.

Take some time to review and compare your work against the model answer. This exercise will help solidify your understanding and provide an opportunity for reflection on how to apply these concepts in future projects.

In the same email, you will also receive a link to a survey, which you can use to self-assess your submission. Once you’ve done that, feel free to progress to the next task.





Auto-graded task

Answer the following in a document titled **machine_learning**.

1. For each of the following examples describe at least one possible **input and output**. Justify your answers:
 - 1.1. A self-driving car
 - 1.2. Netflix recommendation system
 - 1.3. Signature recognition
 - 1.4. Medical diagnosis
2. For each of the following case studies, determine whether it is appropriate to utilise **regression** or **classification** machine learning algorithms. Justify your answers:
 - 2.1. Classifying emails as promotional or social based on their content and metadata.
 - 2.2. Forecasting the stock price of a company based on historical data and market trends.
 - 2.3. Sorting images of animals into different species based on their visual features.
 - 2.4. Predicting the likelihood of a patient having a particular disease based on medical history and diagnostic test results.
3. For each of the following real-world problems, determine whether it is appropriate to utilise a **supervised** or **unsupervised** machine learning algorithm. Justify your answers:
 - 3.1. Detecting anomalies in a manufacturing process using sensor data without prior knowledge of specific anomaly patterns.
 - 3.2. Predicting customer lifetime value based on historical transaction data and customer demographics.
 - 3.3. Segmenting customer demographics based on their purchase history, browsing behaviour, and preferences.
 - 3.4. Analysing social media posts to categorise them into different themes.

4. For each of the following real-world problems, determine whether it is appropriate or inappropriate to utilise **semi-supervised** machine learning algorithms. Justify your answers:
 - 4.1. Predicting fraudulent financial transactions using a dataset where most transactions are labelled as fraudulent or legitimate.
 - 4.2. Analysing customer satisfaction surveys where only a small portion of the data is labelled with satisfaction ratings.
 - 4.3. Identifying spam emails in a dataset where the majority of emails are labelled.
 - 4.4. Predicting the probability of default for credit card applicants based on their complete financial and credit-related information.
5. Export your document to PDF before submitting it.

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.



Share your thoughts

Please take some time to complete this short feedback [form](#) to help us ensure we provide you with the best possible learning experience.
