**Supervised, Semi-Supervised, Unsupervised, and Self-Supervised Learning**

**Demystifying each learning task**

The exponential number of research and publications have introduced many terms and concepts in the domain of machine learning, yet many have degenerated to merely buzzwords without many people fully understanding their differences.

This article demystifies the four core regimes in the field of machine learning — supervised, semi-supervised, unsupervised, and self-supervised learning — and discusses several examples/methods in solving these problems. Enjoy!

**Supervised Learning**

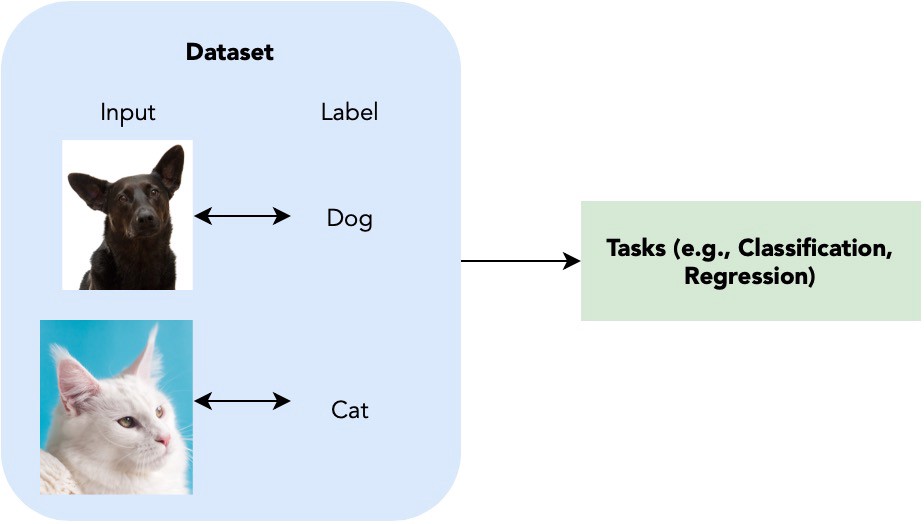


Figure 1. Illustration of Supervised Learning. Image made by author with resources from [Unsplash](https://unsplash.com).

The most common, and perhaps THE type that we refer to when talking about machine learning is supervised learning.

In simple words, **supervised learning** provides a set of input-output pairs such that we can learn an intermediate system that maps inputs to correct outputs.

A naive example of supervised learning is determining the class (CLASSIFICATION) (i.e., dogs/cats, etc) of an image based on a dataset of images and their corresponding classes, which we will refer to as their labels.

With the given input-label pair, the current popular approach will be to directly train a deep neural network (i.e., a convolutional neural network) to output a label prediction from the given image, compute a differentiable loss between the prediction and the actual correct answers, and backpropagate through the network to update weights to optimise the predictions.

Overall, supervised learning is the most straightforward type of learning method as it assumes the labels of each image is given, which eases up the process of learning as it is easier for the network to learn.

**Semi-Supervised Learning**

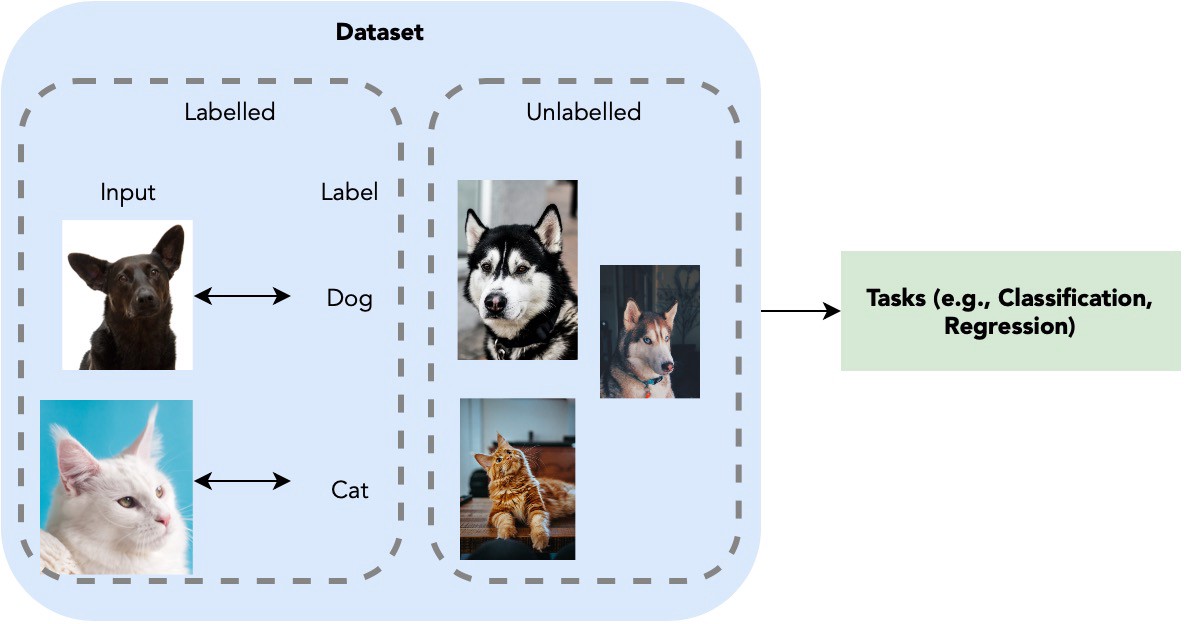


Figure 2. Illustration of Semi-upervised Learning. Image made by author with resources from [Unsplash](https://unsplash.com).

While supervised learning assumes the entire dataset to be trained on a task has the corresponding labels for each input, reality may not always be like this. Labelling is a labour-intensive processing task and often input data comes in unpaired.

**Semi-supervised learning** aims to address this problem: how do we use a small set of input-output pairs and another set of only inputs to optimise a model for a task that we are solving?

Referring back to the image classification task, image and the image labels now only exist partially within the dataset. Is it possible to still utilise the data without any labelling?

Short answer, yes. In fact, there is an easy trick called pseudo-labelling to do this. First, we use the images with correct labels to train a classification model. We then use this classification model to label the unlabelled images. Images with labels of high confidence from model will then be added to the model with their predicted labels as pseudo-labels for continued training. We iterate this process until all the data are utilised for the best classification model.

Of course, this method while seemingly smart, may easily go wrong. If the number of labelled data is very limited, it is very likely the model overfits on the training data and give false pseudo-labels at an early stage, leading to the entire model being completely wrong. It is thus also very important to decide the confidence threshold to include a input-pseudo-label pair into training.

To avoid the model overfitting at an early stage, one could also adopt data augmentations techniques to increase the size of training and creating a wider distribution of data. *If interested, you may also refer to my article on* [*mixup*](https://towardsdatascience.com/enhancing-neural-networks-with-mixup-in-pytorch-5129d261bc4a) *as one of the most predominant augmentation strategy for image classification tasks.*

**Unsupervised Learning**

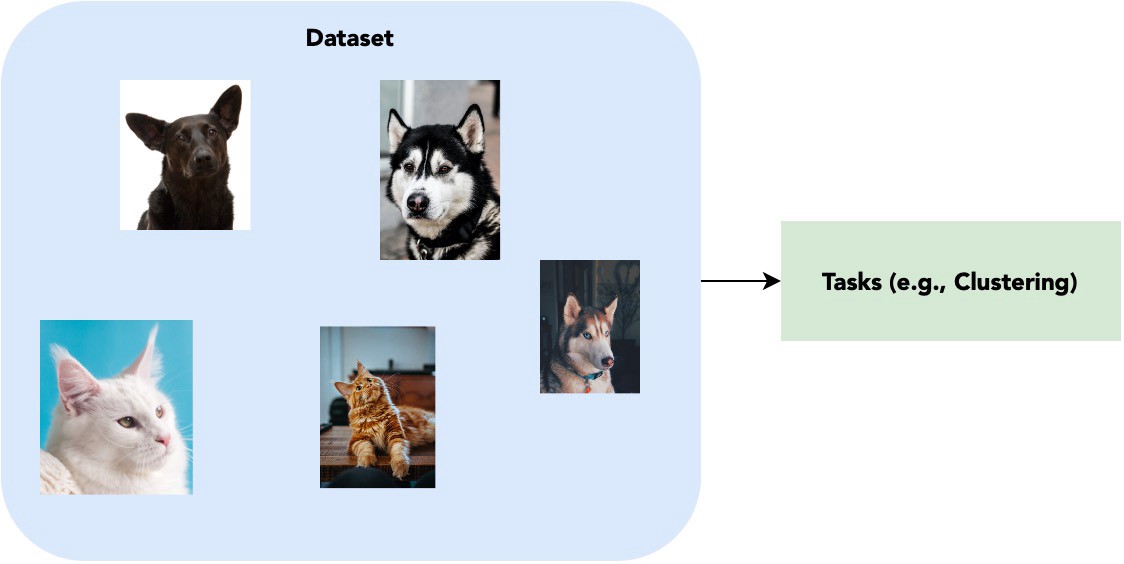


Figure 3. Illustration of Unsupervised Learning. Image made by author with resources from [Unsplash](https://unsplash.com).

Now that we understand how to use minimal labels for training, we can think one step further: a dataset with no labels at all.

**Unsupervised learning** is the at other end of the spectrum, where only input data have no corresponding classifications or labelling. The goal is to find underlying patterns with each dataset.

Tasks involving unsupervised learning include customer segmentation, recommendation systems, and many more. However, how does one learn anything without any labels?

Since we have no ‘correct answer’ to each input label, the best way we can somehow find a pattern is to cluster them. That is, given a set of data features, we try to find features that are similar to each other and group them together. Some clustering methods include K-Means and K-Medoids methods.

Merely clustering may actually create numerous insights. Take the example of a recommendation system: by grouping the users based on their activities, one can recommend contents one user favours to the other without explicitly understanding what the interests of each user is.

**Self-Supervised Learning**

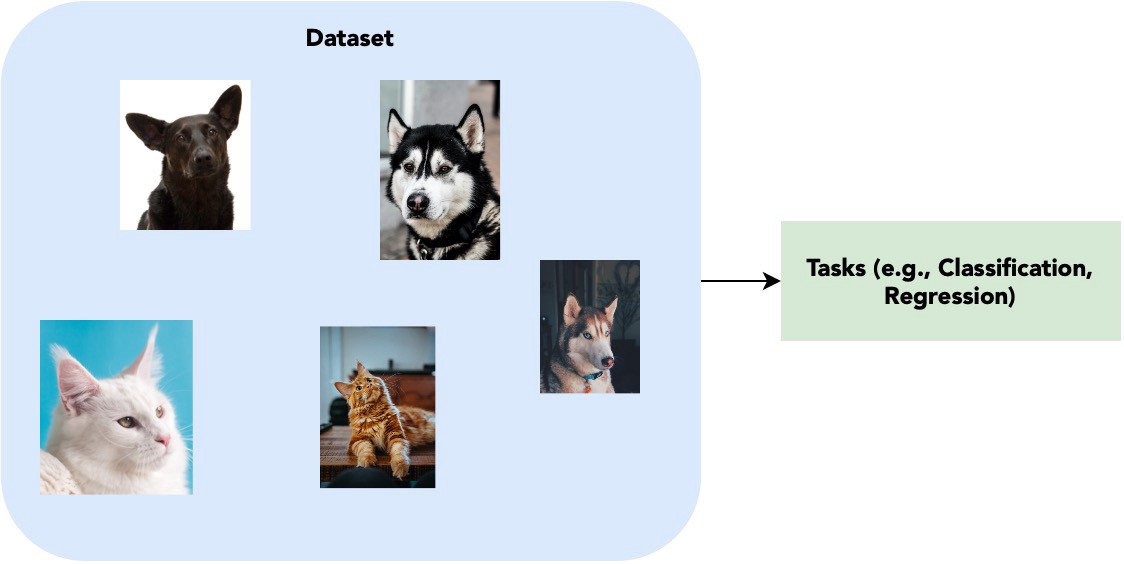


Figure 4. Illustration of Self-Supervised Learning. Image made by author with resources from [Unsplash](https://unsplash.com). Self-supervised learning is very similar to unsupervised, except for the fact that self-supervised learning aims to tackle tasks that are traditionally done by supervised learning.

Now comes to the tricky bit. It seems like we have covered the entire spectrum of learning, then what in the world is self-supervised learning!? Well, the answer may be simpler than you think!

**Self-supervised learning** is in some sense a type of unsupervised learning as it follows the criteria that no labels were given. However, instead of finding high-level patterns for clustering, self-supervised learning attempts to still solve tasks that are traditionally targeted by supervised learning (e.g., image classification) without any labelings available.

This may seem impossible at first glance, but numerous recent research have came up with creative and interesting techniques to learn this, one of which being the infamous contrastive learning from positive and negative pairs.

In short, one performs augmentations on the same images and label them as a positive pair, different images as negative pair, and attempts to push the learnt features of negative pairs away while dragging positive features close. This enables the network to learn to group images of similar classes, which further make tasks like classifications and segmentations that originally required fixed labels to learn become possible without given ground truths.

**Testing Each Concept**

If you want to personally experience individual concepts, you may retrieve any datasets and personally remove partial or all labels to test out each learning methods.

A thing to note is that if you directly retrieve the dataset from torchvision, they already have pre-defined the labels for you, and downloading and writing your own dataloader will be better if you want to test out semi-supervised or self-supervised learning. One place that I have found to be particularly useful for retrieving datasets is [Graviti Open Dataset](https://gas.graviti.com/open-datasets). With so many different datasets and papers, it is often tiring to understand what dataset to use. The Open Dataset platform organises all the popular datasets so that you could easily find them and be redirected to their official websites. They are also currently working on providing a good API to help ease up the process of designing data loaders which I believe will be great for future use.

I would personally recommend to test on MNIST or CIFAR-10 datasets that requires less computational power. They would be easier to train if only a laptop is available for you.

**Additional Papers to Read**

For more papers and techniques in self-supervised learning, you may also refer to papers from [paperswithcode](https://paperswithcode.com/task/self-supervised-learning) that includes the current benchmarks and state-of-the-art methods.

**Conclusion**

And there you have it! Hopefully after this article you will know the subtle differences between supervised, semi-supervised, unsupervised, and self-supervised learning. Have fun and keep learning!

# Understanding the Different Types of Machine Learning Models

## Supervised, Semi-supervised, and Unsupervised machine learning methods



Photo by [Franck V.](https://unsplash.com/@franckinjapan?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

# Overview

Within the field of machine learning, there are three main types of tasks: supervised, semi-supervised, and unsupervised.

The main difference between these types is the level of availability of **ground truth** data, which is prior knowledge of what the output of the model should be for a given input.

**Supervised learning** aims to learn a function that, given a sample of data and desired outputs, approximates a function that maps inputs to outputs.

**Semi-supervised learning** aims to label unlabeled data points using knowledge learned from a small number of labeled data points.

**Unsupervised learning** does not have (or need) any labeled outputs, so its goal is to infer the natural structure present within a set of data points.

# Supervised Learning

Supervised learning models map inputs to outputs.

## Overview

Supervised learning is typically done in the context of **classification**, when we want to map input to output labels, or **regression**, when we want to map input to a continuous output. Common algorithms in supervised learning include logistic regression, naive bayes, support vector machines, artificial neural networks, and random forests. In both regression and classification, the goal is to find specific relationships or structure in the input data that allow us to effectively produce correct output data.

Note that “correct” output is determined entirely from the training data, so while we do have a ground truth that our model will assume is true, it is not to say that data labels are always correct in real-world situations. Noisy, or incorrect, data labels will clearly reduce the effectiveness of your model.

## ****Complexity****

Model complexity refers to the complexity of the function you are attempting to learn — similar to the degree of a polynomial. The proper level of model complexity is generally determined by the nature of your training data.

If you have a small amount of data, or if your data is not uniformly spread throughout different possible scenarios, you should opt for a low-complexity model. This is because a high-complexity model will **overfit** if used on a small number of data points.

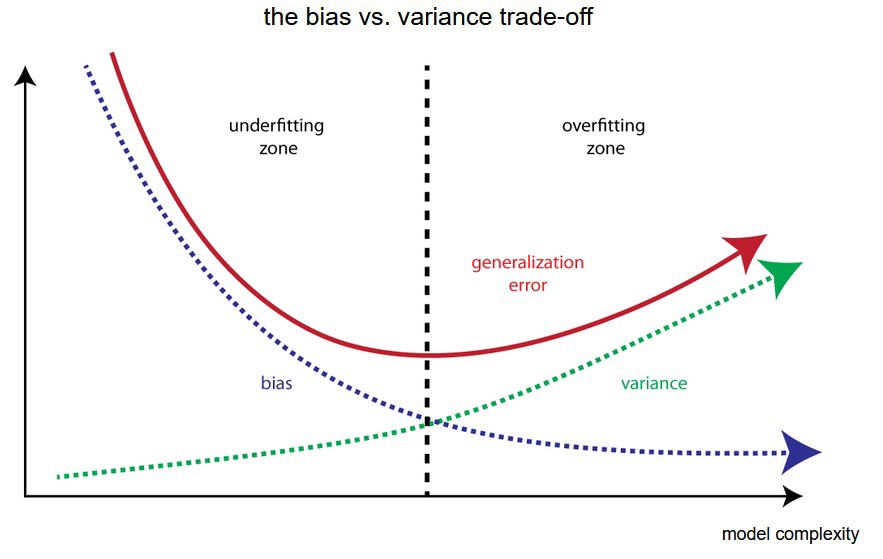
Overfitting refers to learning a function that fits your training data very well, but does not **generalize** to other data points — in other words, you are strictly learning to produce your training data without learning the actual trend or structure in the data that leads to this output. Imagine trying to fit a curve between 2 points. In theory, you can use a function of any degree, but in practice, you would parsimoniously add complexity, and go with a linear function.

## Bias-variance trade-off

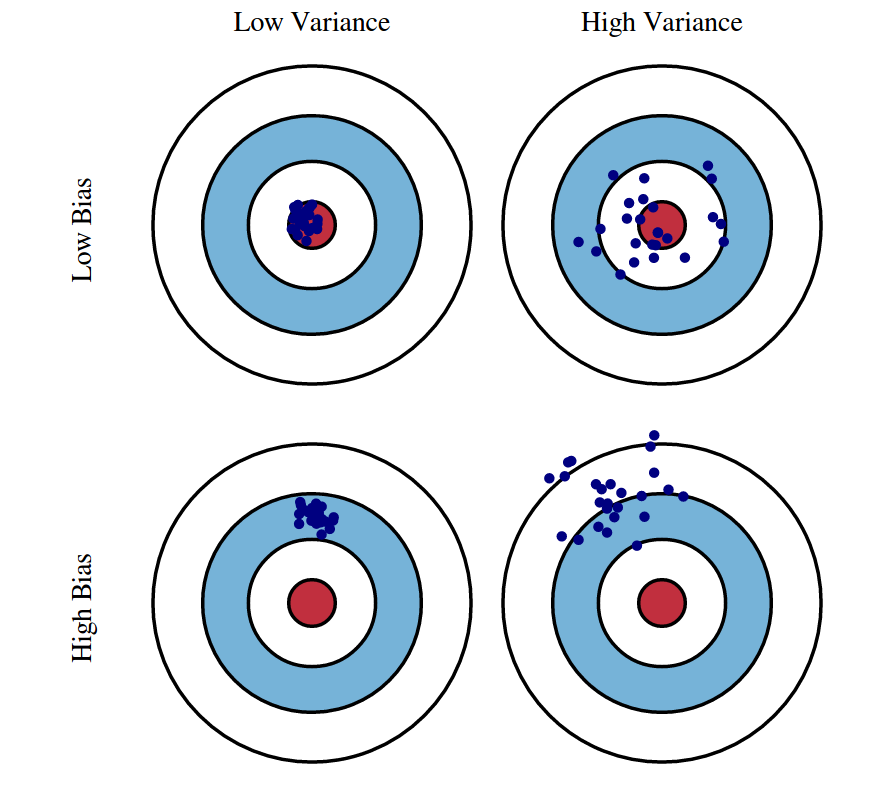
The bias-variance trade-off also relates to model generalization. In any model, there is a balance between **bias**, which is the constant error term, and **variance**, which is the amount by which the error may vary between different data sets. So, high bias and low variance would be a model that is consistently wrong 20% of the time, whereas a low bias and high variance model would be a model that can be wrong anywhere from 5%-50% of the time, depending on the data used to train it.

Note that bias and variance typically move in opposite directions of each other; increasing bias will usually lead to lower variance, and vice versa. When making your model, your specific problem and the nature of your data should allow you to make an informed decision on where to fall on the bias-variance spectrum.

Generally, increasing bias (and decreasing variance) results in models with relatively guaranteed baseline levels of performance, which may be critical in certain tasks. Additionally, in order to produce models that generalize well, the variance of your model should scale with the size and complexity of your training data. Small, simple data-sets should usually be learned with low-variance models, and large, complex data-sets will often require higher-variance models to fully learn the structure of the data.







# Semi-supervised Learning

Learning with both unlabaled and labeled data points.

## Overview

Semi-supervised learning falls in between supervised and unsupervised learning. Semi-supervised models aim to use a small amount of labeled training data along with a large amount of unlabeled training data. This often occurs in real-world situations in which labeling data is very expensive, and/or you have a constant stream of data.

For example, if we were trying to detect inappropriate messages in a social network, there is no way to obtain hand-labeled information on each message, as there are simply too many and it would be too costly. Instead, we can hand-label a subset of them, and leverage semi-supervised techniques to use this small set of labeled data to help us understand the rest of the messages’ content as they come in.

Some common semi-supervised methods are transductive support vector machines, and graph-based methods such as label propagation.

## Assumptions

Semi-supervised methods must make some assumption about the data in order to justify using a small set of labeled data to make conclusions about the unlabeled data points. These can be grouped into three categories.

The first is the **continuity assumption**. This assumes that data points that are “close” to each other are more likely to have a common label.

The second is the **cluster assumption**. This assumes that the data naturally forms discrete clusters, and that points in the same cluster are more likely to share a label.

The third is the **manifold assumption**. This assumes that the data roughly lies in a lower-dimensional space (or manifold) than the input space. This scenario is relevant when an unobservable or difficult-to-observe system with a small number of parameters produces high-dimensional observable output.

# Unsupervised Learning

Unsupervised models find inherent patterns in data.

## Overview

The most common tasks within unsupervised learning are clustering, representation learning, and density estimation. In all of these cases, we wish to learn the inherent structure of our data without using explicitly-provided labels. Some common algorithms include k-means clustering, principal component analysis, and autoencoders. Since no labels are provided, there is no specific way to compare model performance in most unsupervised learning methods.

## Exploratory data analysis (EDA)

Unsupervised learning is very useful in exploratory analysis because it can automatically identify structure in data. For example, if an analyst were trying to segment consumers, unsupervised clustering methods would be a great starting point for their analysis. In situations where it is either impossible or impractical for a human to propose trends in the data, unsupervised learning can provide initial insights that can then be used to test individual hypotheses.

## Dimensionality reduction

Dimensionality reduction, which refers to the methods used to represent data using less columns or features, can be accomplished through unsupervised methods. In **representation learning**, we wish to learn relationships between individual features, allowing us to represent our data using the latent features that interrelate our initial features. This sparse latent structure is often represented using far fewer features than we started with, so it can make further data processing much less intensive, and can eliminate redundant features. In other contexts, dimensionality reduction may be used to convert data from one modality to another. For example, a recurrent autoeconder may be used to convert sequences into a fixed length representation.

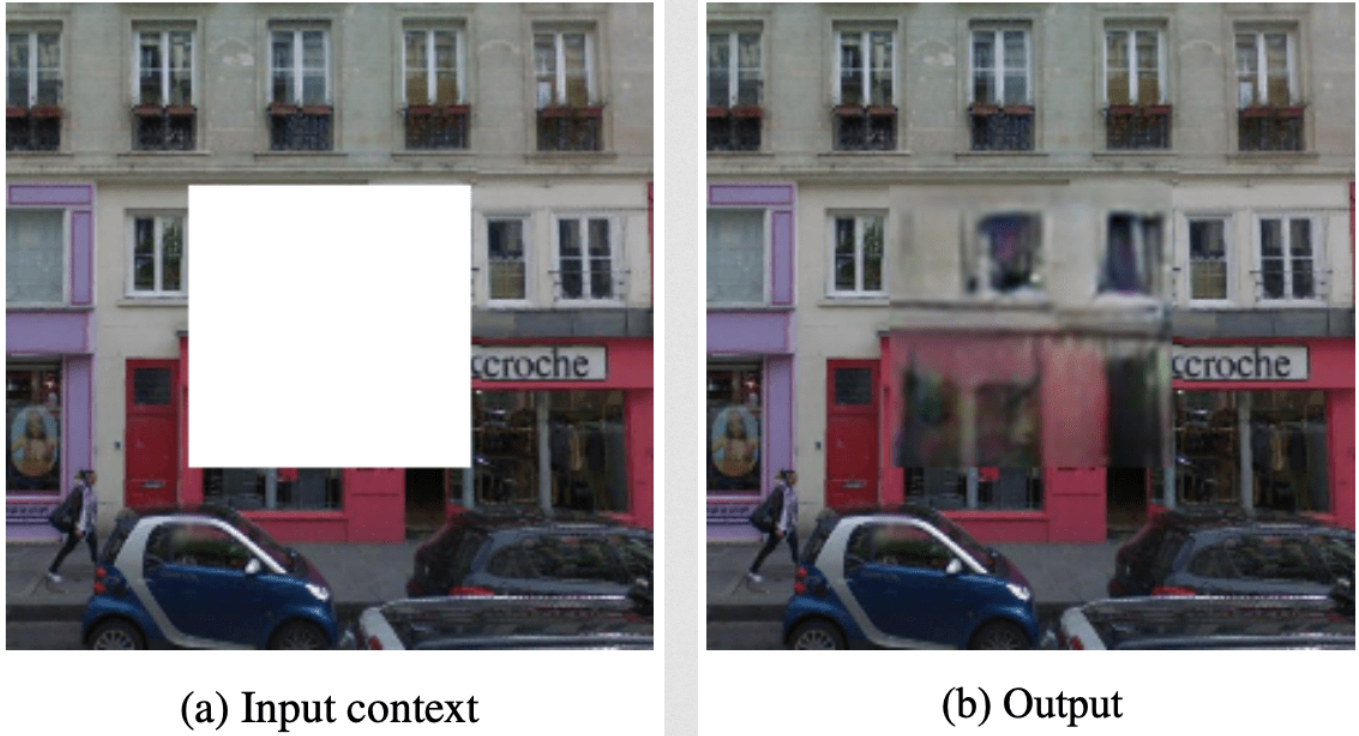
# In-Depth Guide to Self-Supervised Learning: Benefits & Uses

Supervised learning has been a popular set of machine learning techniques that work effectively in performing regression and classification tasks. However, supervised learning models require [manual data labeling](https://research.aimultiple.com/data-labeling/) which slows down the model building process, is expensive, and error prone.

Self-supervised learning (SSL), also known as self-supervision, is an emerging solution to the challenge posed by data labeling. By building models autonomously, self-supervised learning reduces the cost and time to build machine learning models. In this article, we dive into self-supervised learning and compare it with other machine learning approaches such as supervised and unsupervised learning.

## ****What is self-supervised learning?****

Self-supervised learning is a machine learning approach where the model trains itself by leveraging one part of the data to predict the other part and generate labels accurately. In the end, this learning method converts an unsupervised learning problem into a supervised one. Below is an example of a self-supervised learning output.



Source: [Arxiv](https://arxiv.org/pdf/1604.07379.pdf)

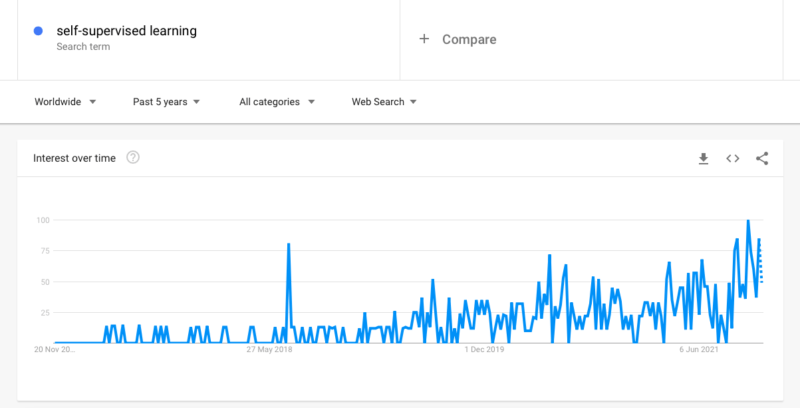
## Why is self-supervised learning important now?

Most machine learning techniques require training datasets to make predictions. Data scientists need to label the observations in the training datasets manually or with [data labeling tools](https://research.aimultiple.com/data-labeling-tools/) to enable AI to understand the input data and make accurate predictions about new data. In cases where the training dataset is too large, manually labeling training data can be quite costly and time-consuming.

Self-supervised learning eliminates the necessity of data labeling. It enables computers to label, categorize, and analyze data themselves.

## What is the level of interest in self-supervised learning?

As seen in the graph below, there is a steady increase in the level of interest in self-supervised learning since researchers from Google introduced the [BERT](https://arxiv.org/pdf/1810.04805.pdf) model at the end of 2018 which leverages self-supervised learning for [natural language processing (NLP)](https://research.aimultiple.com/nlp/) tasks.



Source: [Google Trends](https://trends.google.com/trends/explore?date=today%205-y&q=self-supervised%20learning)

Since AI/ML models require huge datasets and labeling this data is one of the biggest [challenges](https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/what-ai-can-and-cant-do-yet-for-your-business) of machine learning adoption, we expect this trend to continue.

## What are its differences from supervised/unsupervised learning?

### Supervised learning vs self-supervised learning

The common characteristic of supervised and self-supervised learning is that both methods build learning models from training datasets with their labels. However, self-supervised learning doesn’t require manual labeling since it generates them by itself.

### Semi-supervised learning vs self-supervised learning

Semi-supervised learning uses manually labeled training data for supervised learning and unsupervised learning approaches for unlabeled data to generate a model that leverages existing labels but builds a model that can make predictions beyond the labeled data. Self-supervised learning relies completely on data that lacks manually generated labels.

### Unsupervised learning vs self-supervised learning

Self-supervised learning is similar to unsupervised learning because both techniques work with datasets that don’t have manually added labels. Accordingly, self-supervised learning can be considered as a subset of unsupervised learning. However, unsupervised learning concentrates on clustering, grouping, and dimensionality reduction, while self-supervised learning aims to draw conclusions for regression and classification tasks.

### Hybrid approaches vs self-supervised learning

There are also hybrid approaches that combine automated data labeling tools with supervised learning. In such methods, computers can label data points that are easier to label by relying on their training data and leave the complex ones to humans. Or, they can label all data points automatically but need human approval.



In self-supervised learning, automated data labeling is embedded in the training model. The dataset is labeled as part of the learning processes; thus, it doesn’t ask for human approval or only label the simple data points.

## Why do we need self-supervised learning?

### Scalability

Supervised learning requires labeled data to predict outcomes for unknown data. However, it can need large datasets to build proper models and make accurate predictions. For large training datasets, manual data labeling can be challenging. Self-supervised learning can automate this process and handle this task with even massive amounts of data.

### Improved AI capabilities

Today, self-supervised learning is mostly used in [computer vision](https://research.aimultiple.com/computer-vision/) for tasks like colorization, 3D rotation, depth completion, or context filling. These tasks require example labeled cases to build accurate models but self-supervised learning can improve computer vision or speech recognition technologies by eliminating the necessity of example cases.

### Understanding how the human mind works

Supervised models require human intervention to perform appropriately. However, those interventions don’t always exist. Then, we can think of introducing reinforcement learning to machines to make them start from the beginning in cases where they can get immediate feedback without negative consequences. However, this does not cover many real-world scenarios. Humans can think through the consequences of their actions before making them, and they don’t have to experience all actions to decide on what to do. Machines also have the potential to work in the same way.

Self-supervised learning steps in at this point. It automatically generates labels without human intervention and enables machines to come up with a solution without any interference. Facebook VP and chief AI scientist Yann LeCun [shares](https://venturebeat.com/2020/05/02/yann-lecun-and-yoshua-bengio-self-supervised-learning-is-the-key-to-human-level-intelligence/) that self-supervised learning is a step towards how human intelligence works. As we understand this better, we will get closer to create models that think more similar to humans.

## What are its applications?

Self-supervised learning technologies mostly focus on improving computer vision and natural language processing (NLP) capabilities.

* **Colorization:**SSL can be used for coloring grayscale images, as seen below. [](https://research.aimultiple.com/wp-content/uploads/2020/06/Colorization_SSL.jpg)Source: [Perfectial](https://perfectial.com/blog/self-supervised-learning/)
* **Context Filling:** SSL can fill a space in an image or predict a gap in a voice recording or a text.
* **Video Motion Prediction:** Self-supervised learning can provide a distribution of all possible video frames after a specific frame.

Other use cases include:

* [Healthcare](https://research.aimultiple.com/healthcare-ai/)**:** Self-supervised learning can help robotic surgeries perform better by estimating dense depth in the human body. It can also provide better medical visuals with improved computer vision technologies such as colorization and context filling.
* [Autonomous driving](https://research.aimultiple.com/aut/)**:** SSL can be used in estimating the roughness of the terrain. It can also be useful for depth completion to identify the distance to the other cars, people, or other objects while driving.
* [Chatbots](https://research.aimultiple.com/chatbot/)**:** Self-supervised systems can also be applied to chatbots. [Transformers](https://thenextweb.com/neural/2020/04/05/self-supervised-learning-is-the-future-of-ai-syndication/), a chatbot that leverages self-supervised learning, is successful in processing words and mathematical symbols easily. However, it is still far from understanding human language.

## What are its limitations?

* **Can be computationally intense**: Learning models with labels can be built much faster compared to unlabeled learning models. Plus, self-supervised learning autonomously generates labels for the given dataset, which is an additional task. Therefore, compared to supervised learning methods, self-supervised learning can demand more computing power.
* **Labeling accuracy**: You always achieve the best results when you already have labels for your dataset. Self-supervised learning is a solution for when you don’t have any and need to generate them manually. However, the model can come up with inaccurate labels while processing and those inaccuracies can lead to inaccurate results for your task. Thus, labeling accuracy is an additional factor to consider about self-supervised models.

## To learn more on self-supervised learning

[Yann LeCun](http://yann.lecun.com/), VP and Chief AI Scientist at Facebook, is explaining how self-supervised learning works. You can watch the video of his lesson at New York University to learn more about the technical details of this approach.