Semi-Supervised Learning

***Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms. It uses the combination of labeled and unlabeled data sets during the training period.***

Before understanding the Semi-Supervised learning, you should know the main categories of [Machine Learning](https://www.javatpoint.com/machine-learning) algorithms. Machine Learning consists of three main categories: [**Supervised Learning**](https://www.javatpoint.com/supervised-machine-learning)**,**[**Unsupervised Learning**](https://www.javatpoint.com/unsupervised-machine-learning)**, and**[**Reinforcement Learning**](https://www.javatpoint.com/reinforcement-learning)**.** Further, the basic difference between Supervised and unsupervised learning is that *supervised learning datasets consist of an output label training data associated with each tuple,* and *unsupervised datasets do not consist the same.****Semi-supervised learning is an important category that lies between the Supervised and Unsupervised machine learning.*** Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for the corporate purpose, it may have few labels.

The basic disadvantage of supervised learning is that it requires hand-labeling by ML specialists or data scientists, and it also requires a high cost to process. Further unsupervised learning also has a limited spectrum for its applications. **To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced**. In this algorithm, training data is a combination of both labeled and unlabeled data. However, labeled data exists with a very small amount while it consists of a huge amount of unlabeled data. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data. It is why label data is a comparatively, more expensive acquisition than unlabeled data.

We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college. Further, if that student is self-analyzing the same concept without any help from the instructor, it comes under unsupervised learning. Under semi-supervised learning, the student has to revise itself after analyzing the same concept under the guidance of an instructor at college.

Assumptions followed by Semi-Supervised Learning

To work with the unlabeled dataset, there must be a relationship between the objects. To understand this, semi-supervised learning uses any of the following assumptions:

* **Continuity Assumption:** As per the continuity assumption, the objects near each other tend to share the same group or label. This assumption is also used in supervised learning, and the datasets are separated by the decision boundaries. But in semi-supervised, the decision boundaries are added with the smoothness assumption in low-density boundaries.
* **Cluster assumptions-** In this assumption, data are divided into different discrete clusters. Further, the points in the same cluster share the output label.
* **Manifold assumptions-** This assumption helps to use distances and densities, and this data lie on a manifold of fewer dimensions than input space.
* The dimensional data are created by a process that has less degree of freedom and may be hard to model directly. **(This assumption becomes practical if high).**

Working of Semi-Supervised Learning

Semi-supervised learning uses pseudo labeling to train the model with less labeled training data than supervised learning. The process can combine various neural network models and training ways. The whole working of semi-supervised learning is explained in the below points:

* Firstly, it trains the model with less amount of training data similar to the supervised learning models. The training continues until the model gives accurate results.
* The algorithms use the unlabeled dataset with pseudo labels in the next step, and now the result may not be accurate.
* Now, the labels from labeled training data and pseudo labels data are linked together.
* The input data in labeled training data and unlabeled training data are also linked.
* In the end, again train the model with the new combined input as did in the first step. It will reduce errors and improve the accuracy of the model.

# Semi-Supervised Learning, Explained with Examples

As it sometimes happens, when one approach doesn’t work to solve a problem, you try a different one. When that approach doesn’t work either, it may be a good idea to combine the best parts of both. At least that’s often the case with technology tasks. And [machine learning](https://www.altexsoft.com/whitepapers/machine-learning-bridging-between-business-and-data-science/) is no exception. You’ve probably heard of the two main ML techniques — supervised and unsupervised learning. The marriage of both those technologies gave birth to the happy medium known as *semi-supervised* learning.

[**Supervised learning**](https://www.altexsoft.com/blog/business/supervised-learning-use-cases-low-hanging-fruit-in-data-science-for-businesses/) is training a machine learning model using the labeled dataset. Organic labels are often available in data, but the process may involve a human expert that adds tags to raw data to show a model the target attributes (answers). In simple terms, a label is basically a description showing a model what it is expected to predict.

Supervised learning has a few limitations. This process is

* *slow* (it requires human experts to manually [label training examples](https://www.altexsoft.com/blog/datascience/how-to-organize-data-labeling-for-machine-learning-approaches-and-tools/) one by one) and
* *costly* (a model should be trained on the large volumes of hand-labeled data to provide accurate predictions).

[**Unsupervised learning**](https://www.altexsoft.com/blog/unsupervised-machine-learning/), on the other hand, is when a model tries to mine hidden patterns, differences, and similarities in unlabeled data by itself, without human supervision. Hence the name. Within this method, data points are grouped into clusters based on similarities.

While unsupervised learning is a cheaper way to perform training tasks, it isn’t a silver bullet. Commonly, the scenario

* has a *limited area of applications* (mostly for clustering purposes) and
* provides *less accurate results*.

**Semi-supervised learning** bridges supervised learning and unsupervised learning techniques to solve their key challenges. With it, you train an initial model on a few labeled samples and then iteratively apply it to the greater number of unlabeled data.

* Unlike unsupervised learning, SSL works for a variety of problems from classification and regression to clustering and association.
* Unlike supervised learning, the method uses small amounts of labeled data and also large amounts of unlabeled data, which reduces expenses on manual annotation and cuts [data preparation](https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/) time.

*Speaking of supervised learning, we have an informed 14-min video explaining how data is prepared for it. Make sure you check it out.*

Since unlabeled data is abundant, easy to get, and cheap, semi-supervised learning finds many applications, while the accuracy of results doesn’t suffer.

Let’s look at one of the real-world scenarios like [fraud detection](https://www.altexsoft.com/whitepapers/fraud-detection-how-machine-learning-systems-help-reveal-scams-in-fintech-healthcare-and-ecommerce/). Say, a company with 10 million users analyzed five percent of all transactions to classify them as fraudulent or not while the rest of the data wasn’t labeled with “fraud” and “non-fraud” tags. In this case, semi-supervised learning allows for running all of the information without having to hire an army of annotators or sacrifice accuracy. Below, we’ll explain how exactly this magic works.

## How semi-supervised learning works

Imagine, you have collected a large set of unlabeled data that you want to train a model on. Manual labeling of all this information will probably cost you a fortune, besides taking months to complete the annotations. That’s when the semi-supervised machine learning method comes to the rescue.

The working principle is quite simple. Instead of adding tags to the entire dataset, you go through and hand-label just a small part of the data and use it to train a model, which then is applied to the ocean of unlabeled data.

### Self-training

One of the simplest examples of semi-supervised learning, in general, is self-training.

**Self-training** is the procedure in which you can take any supervised method for classification or regression and modify it to work in a semi-supervised manner, taking advantage of labeled and unlabeled data. The standard workflow is as follows.

*Semi-supervised self-training method*

* You pick a small amount of labeled data, e.g., images showing cats and dogs with their respective tags, and you use this dataset to train a base model with the help of ordinary supervised methods.
* Then you apply the process known as *pseudo-labeling* — when you take the partially trained model and use it to make predictions for the rest of the database which is yet unlabeled. The labels generated thereafter are called *pseudo* as they are produced based on the originally labeled data that has limitations (say, there may be an uneven representation of classes in the set resulting in bias — more dogs than cats).
* From this point, you take the most confident predictions made with your model (for example, you want the confidence of over 80 percent that a certain image shows a cat, not a dog). If any of the pseudo-labels exceed this confidence level, you add them into the labeled dataset and create a new, combined input to train an improved model.
* The process can go through several iterations (10 is often a standard amount) with more and more pseudo-labels being added every time. Provided the data is suitable for the process, the performance of the model will keep increasing at each iteration.

While there are successful examples of self-training being used, it should be stressed that the performance may vary a lot from one dataset to another. And there are plenty of cases when self-training may decrease the performance compared to taking the supervised route.

### Co-training

Derived from the self-training approach and being its improved version, **co-training** is another semi-supervised learning technique used when only a small portion of labeled data is available. Unlike the typical process, co-training trains two individual classifiers based on two *views* of data.

The views are basically different sets of features that provide additional information about each instance, meaning they are independent given the class. Also, each view is sufficient — the class of sample data can be accurately predicted from each set of features alone.

The original [co-training research paper](https://www.cs.cmu.edu/~avrim/Papers/cotrain.pdf) claims that the approach can be successfully used, for example, for web content classification tasks. The description of each web page can be divided into two views: one with words occurring on that page and the other with anchor words in the link leading to it.

*Semi-supervised co-training method*

So, here is how co-training works in simple terms.

* First, you train a separate classifier (model) for each view with the help of a small amount of labeled data.
* Then the bigger pool of unlabeled data is added to receive pseudo-labels.
* Classifiers co-train one another using pseudo-labels with the highest confidence level. If the first classifier confidently predicts the genuine label for a data sample while the other one makes a prediction error, then the data with the confident pseudo-labels assigned by the first classifier updates the second classifier and vice-versa.
* The final step involves the combining of the predictions from the two updated classifiers to get one classification result.

As with self-training, co-training goes through many iterations to construct an additional training labeled dataset from the vast amounts of unlabeled data.

### SSL with graph-based label propagation

A popular way to run SSL is to represent labeled and unlabeled data in the form of graphs and then apply a [**label propagation algorithm**](https://pages.cs.wisc.edu/~jerryzhu/pub/CMU-CALD-02-107.pdf). It spreads human-made annotations through the whole data network.

*A typical example of label propagation*

If you look at the graph, you will see a network of data points, most of which are unlabeled with four carrying labels (two red points and two green points to represent different classes). The task is to spread these colored labels throughout the network. One way of doing this is you pick, say, point 4, and count up all the different paths that travel through the network from 4 to each colored node. If you do that, you will find that there are five walks leading to red points and only four walks leading to green ones. From that, we can assume that point 4 belongs to the red category. And then you will repeat this process for every point on the graph.

The practical use of this method can be seen in personalization and recommender systems. With label propagation, you can predict customer interests based on the information about other customers. Here, we can apply the variation of continuity assumption — if two people are connected on social media, for example, it’s highly likely that they will share similar interests.

## Semi-supervised learning examples

With the amount of data constantly growing by leaps and bounds, there’s no way for it to be labeled in a timely fashion. Think of an active TikTok user that uploads up to [20 videos per day](https://dataprot.net/statistics/tiktok-statistics/) on average. And there are 1 billion active users. In such a scenario, semi-supervised learning can boast of a wide array of use cases from image and speech recognition to web content and text document classification.

### Speech recognition

Labeling audio is a very resource- and time-intensive task, so semi-supervised learning can be used to overcome the challenges and provide better performance. Facebook (now Meta) has [successfully applied](https://www.researchgate.net/publication/341084510_Self-Training_for_End-to-End_Speech_Recognition) semi-supervised learning (namely the self-training method) to its speech recognition models and improved them. They started off with the base model that was trained with 100 hours of human-annotated audio data. Then 500 hours of unlabeled speech data was added and self-training was used to increase the performance of the models. As far as the results, the word error rate (WER) decreased by 33.9 percent, which is a significant improvement.

### Web content classification

With billions of websites presenting all sorts of content out there, classification would take a huge team of human resources to organize information on web pages by adding corresponding labels. The variations of semi-supervised learning are used to annotate web content and classify it accordingly to improve user experience. Many search engines, including [Google](https://ai.googleblog.com/2021/07/from-vision-to-language-semi-supervised.html), apply SSL to their ranking component to better understand human language and the relevance of candidate search results to queries. With SSL, Google Search finds content that is most relevant to a particular user query.

### Text document classification

Another example of when semi-supervised learning can be used successfully is in the building of a text [document classifier](https://www.altexsoft.com/blog/document-classification/). Here, the method is effective because it is really difficult for human annotators to read through multiple word-heavy texts to assign a basic label, like a type or genre.

For example, a classifier can be built on top of [deep learning](https://www.altexsoft.com/blog/deep-learning/) neural networks like LSTM (long short-term memory) networks that are capable of finding long-term dependencies in data and retraining past information over time. Usually, training a neural net requires lots of data with and without labels. A semi-supervised learning framework works just fine as you can train a base LSTM model on a few text examples with hand-labeled most relevant words and then apply it to a bigger number of unlabeled samples.

The [SALnet text classifier](https://ojs.aaai.org/index.php/AAAI/article/view/17558/17365) made by researchers from Yonsei University in Seoul, South Korea, demonstrates the effectiveness of the SSL method for tasks like [sentiment analysis](https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/).

## When to use and not use semi-supervised learning

With a minimal amount of labeled data and plenty of unlabeled data, semi-supervised learning shows promising results in classification tasks while leaving the doors open for other ML tasks. Basically, the approach can make use of pretty much any supervised algorithm with some modifications needed. On top of that, SSL fits well for clustering and anomaly detection purposes too if the data fits the profile. While a relatively new field, semi-supervised learning has already proved to be effective in many areas.

But it doesn’t mean that semi-supervised learning is applicable to all tasks. If the portion of labeled data isn’t representative of the entire distribution, the approach may fall short. Say, you need to classify images of colored objects that have different looks from different angles. Unless you have a large amount of labeled data, the results will have poor accuracy. But if we’re talking about lots of labeled data, then semi-supervised learning isn’t the way to go. Like it or not, many real-life applications still need lots of labeled data, so supervised learning won’t go anywhere in the near future.

Difference between Semi-supervised and Reinforcement Learning.

Reinforcement learning is different from semi-supervised learning, as it works with rewards and feedback. ***Reinforcement learning aims to maximize the rewards by their hit and trial actions, whereas in semi-supervised learning, we train the model with a less labeled dataset.***

Real-world applications of Semi-supervised Learning-

Semi-supervised learning models are becoming more popular in the industries. Some of the main applications are as follows.

* **Speech Analysis-** It is the most classic example of semi-supervised learning applications. Since, labeling the audio data is the most impassable task that requires many human resources, this problem can be naturally overcome with the help of applying SSL in a Semi-supervised learning model.
* **Web content classification-** However, this is very critical and impossible to label each page on the internet because it needs mode human intervention. Still, this problem can be reduced through Semi-Supervised learning algorithms.  
  Further, Google also uses semi-supervised learning algorithms to rank a webpage for a given query.
* **Protein sequence classification-** DNA strands are larger, they require active human intervention. So, the rise of the Semi-supervised model has been proximate in this field.
* **Text document classifier-** As we know, it would be very unfeasible to find a large amount of labeled text data, so semi-supervised learning is an ideal model to overcome this.