**Task1**

# How semi-supervised learning works

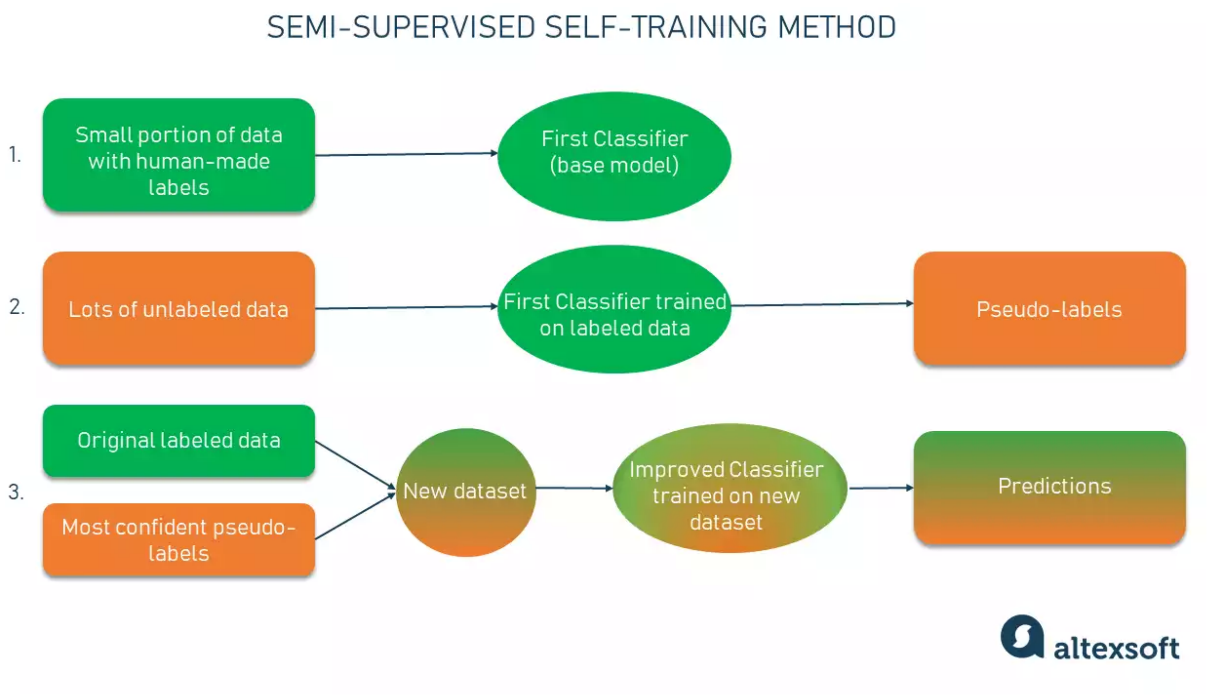
Imagine, you have collected a large set of unlabeled data that you want to train a model on. Manual labeling of all this information will probably cost you a fortune, besides taking months to complete the annotations. That’s when the semi-supervised machine learning method comes to the rescue.

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
1.
Small portion of data with human-made labels
First Classifier (base model)
2.
Lots of unlabeled data
First Classifier trained on labeled data
Pseudo-labels
3.
Original labeled data
Most confident pseudo-labels
New dataset
Improved Classifier trained on new dataset
Predictions
altexsoft

*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.

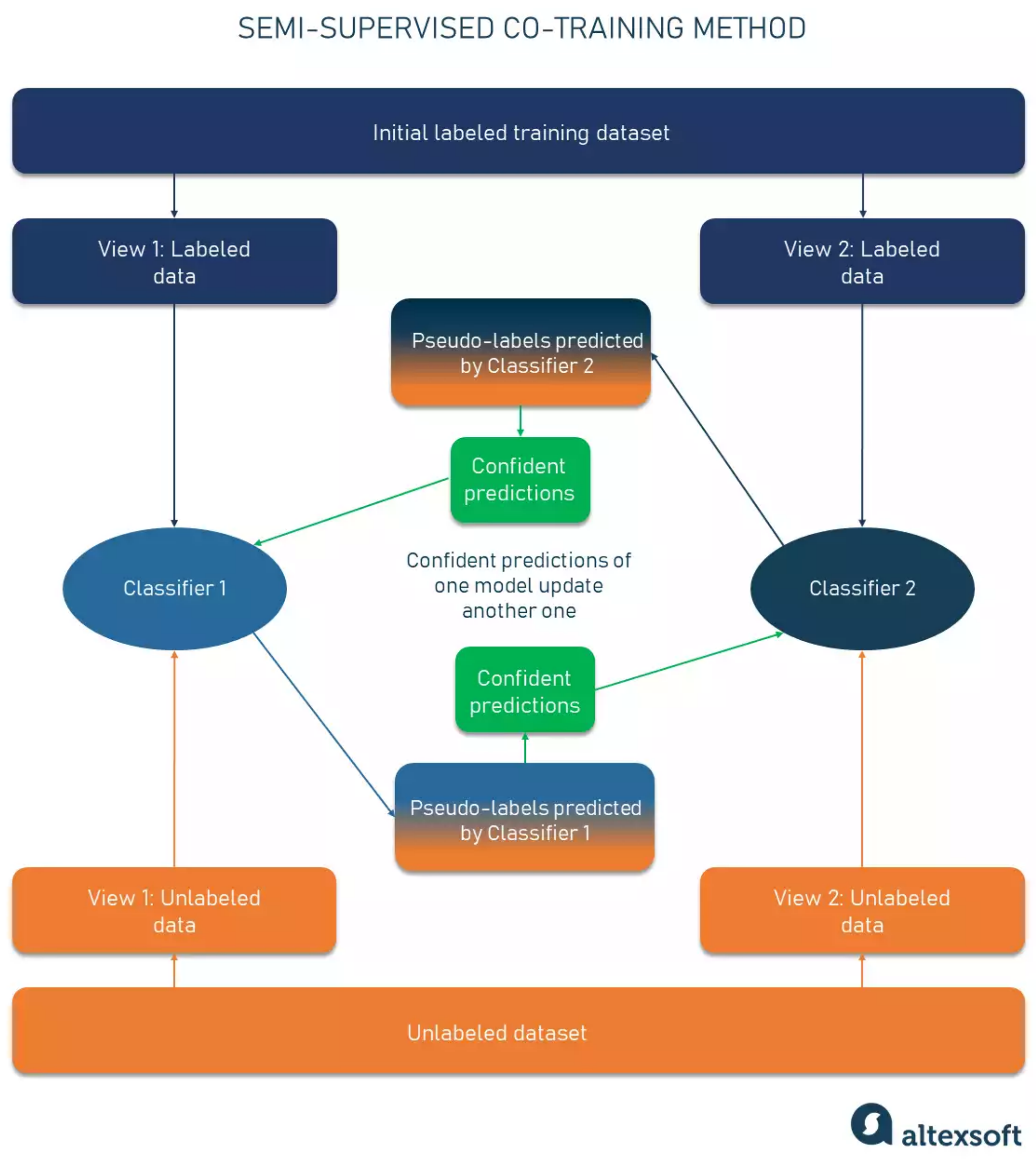
While there are successful examples of self-training being used, it should be stressed that the performance may vary a lot from one dataset to another. And there are plenty of cases when self-training may decrease the performance compared to taking the supervised route.

## 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 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
Initial labeled training dataset
View 1: Labeled data
View 2: Labeled data
Pseudo-labels predicted by Classifier 2
Confident predictions
Classifier 1
Confident predictions of one model update another one
Classifier 2
Confident predictions
Pseudo-labels predicted by Classifier 1
View 1: Unlabeled data
View 2: Unlabeled data
Unlabeled dataset
altexsoft

*Semi-supervised co-training method*

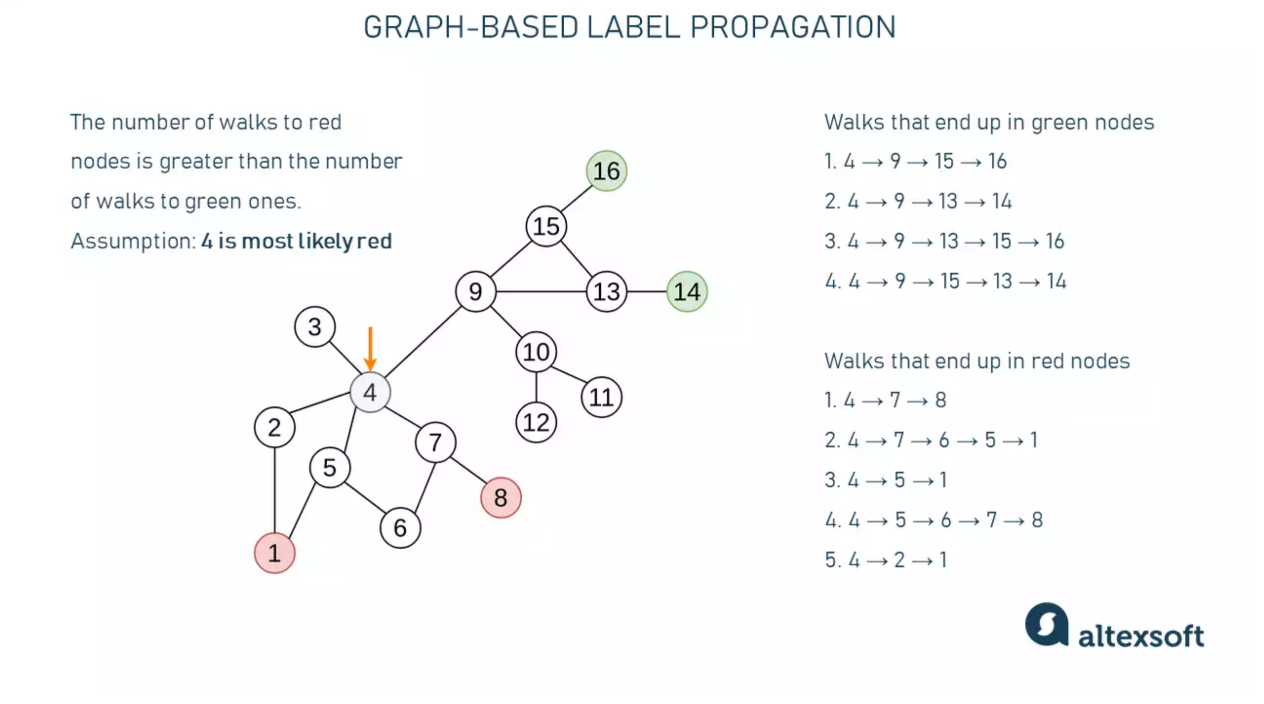
So, here is how co-training works in simple terms.

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**. It spreads human-made annotations through the whole data network.


GRAPH-BASED LABEL PROPAGATION
The number of walks to red nodes is greater than the number of walks to green ones.
Assumption: 4 is most likely red
Walks that end up in green nodes
1. 4 → 9 → 15 → 16
2. 4 → 9 → 13 → 14
3. 4 → 9 → 13 → 15 → 16
4. 4 → 9 → 15 → 13 → 14
Walks that end up in red nodes
1. 4 → 7 → 8
2. 4 → 7 → 6 → 5 → 1
3. 4 → 5 → 1
4. 4 → 5 → 6 → 7 → 8
5. 4 → 2 → 1
altexsoft

*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 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 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, 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. 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 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 made by researchers from Yonsei University in Seoul, South Korea, demonstrates the effectiveness of the SSL method for tasks like sentiment analysis.

## 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.

## Task2
## Non_Statistical_Model_Uses

In machine learning, non-statistical models refer to approaches that are not based on traditional statistical methods. These models often fall under the category of "non-parametric" or "non-linear" techniques, as they do not assume a fixed number of parameters or a specific functional form. Here are some common non-statistical model uses in machine learning:

1. **Neural Networks:** Neural networks are a class of non-linear models inspired by the structure and function of the human brain. They consist of interconnected nodes (neurons) organized into layers, and they are capable of learning complex patterns and representations from data.

2. **Decision Trees:** Decision trees are tree-like structures where each internal node represents a decision based on a feature, and each leaf node represents the outcome or prediction. They are intuitive, easy to interpret, and effective for both classification and regression tasks.

3. **Random Forest:** A random forest is an ensemble learning technique that combines multiple decision trees to make predictions. It improves performance and reduces overfitting compared to a single decision tree.

4. **Gradient Boosting Machines (GBMs):** GBMs are also ensemble methods that build multiple weak learners (usually decision trees) sequentially, with each learner trying to correct the errors of its predecessor. XGBoost and LightGBM are popular implementations of GBMs.

5. **K-Nearest Neighbors (KNN):** KNN is a non-parametric classification algorithm that assigns a new data point's class label based on the majority class of its k-nearest neighbors in the training data.

6. **Support Vector Machines (SVM):** SVM is a powerful supervised learning algorithm that can perform both linear and non-linear classification tasks. It aims to find an optimal hyperplane that maximizes the margin between different classes.

7. **Nearest Centroid Classifier:** This is a simple non-parametric classification algorithm that assigns a new data point to the class with the nearest centroid (mean) in the feature space.

8. **K-Means Clustering:** K-Means is an unsupervised learning algorithm used for clustering data into K distinct groups based on similarity.

9. **Self-Organizing Maps (SOM):** SOM is a type of unsupervised neural network that projects high-dimensional data onto a lower-dimensional space, enabling the visualization and clustering of complex data.

10. **Principal Component Analysis (PCA):** Although not a model in the traditional sense, PCA is a non-linear dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving the most significant variance.

These non-statistical models are widely used in machine learning due to their flexibility, ability to handle complex data, and performance in various tasks. However, the choice of model depends on the specific problem and the characteristics of the data at hand.