Basic concepts and definitions for machine learning

1. What is machine learning?

Machine learning is a set of methods that computers use to make and improve predictions or behaviors based on data.

For example, to predict the value of a house, the computer would learn patterns from past house sales. There are three main types of machine learning:

1. Supervised machine learning, which covers all prediction problems where we have a dataset for which we already know the outcome of interest (e.g., past house prices) and want to learn to predict the outcome for new data. Excluded from supervised learning are for example clustering tasks or unsupervised learning. The goal of supervised learning is to learn a predictive model that maps features of the data (e.g., house size, location, floor type, …) to an output (e.g., house price). If the output is categorical, the task is called classification, and if it is numerical, it is called regression. The machine learning algorithm learns a model by estimating parameters (like weights) or learning structures (like trees). The algorithm is guided by a score or loss function that is minimized. In the house value example, the machine minimizes the difference between the estimated house price and the predicted price. A fully trained machine learning model can then be used to make predictions for new instances.

2. Unsupervised learning: where we do not have a specific outcome of interest but want to find clusters of data points.

Excluded from the previous is ***3. reinforcement learning***, where an agent learns to optimize a certain reward by acting in an environment (e.g., a computer playing Tetris).

Estimation of house prices, product recommendations, street sign detection, credit default prediction and fraud detection: All these examples have in common that they can be solved by machine learning. The tasks are different, but the approach is the same:

> **Step 1**: Data collection. The more, the better. The data must contain the outcome you want to predict and additional information from which to make the prediction. For a street sign detector ("Is there a street sign in the image?"), you would collect street images and label whether a street sign is visible or not. For a credit default predictor, you need past data on actual loans, information on whether the customers were in default with their loans, and data that will help you make predictions, such as income, past credit defaults, and so on. For an automatic house value estimator program, you could collect data from past house sales and information about the real estate such as size, location, and so on.
>
> **Step 2:** Enter this information into a machine learning algorithm that generates a sign detector model, a credit rating model, or a house value estimator.
>
> **Step 3:** Use model with new data. Integrate the model into a product or process, such as a self-driving car, a credit application process, or a real estate marketplace website.

Machines ***surpass*** humans in many tasks, such as playing chess (or more recently Go) or predicting the weather. Even if the machine is as good as a human or a bit worse at a task, there remain great advantages in terms of speed, reproducibility, and scaling. A once implemented machine learning model can complete a task much faster than humans, reliably delivers consistent results and can be copied infinitely. Replicating a machine learning model on another machine is fast and cheap. The training of a human for a task can take decades (especially when they are young) and is very costly. A major disadvantage of using machine learning is that insights about the data and the task the machine solves is hidden in increasingly complex models.

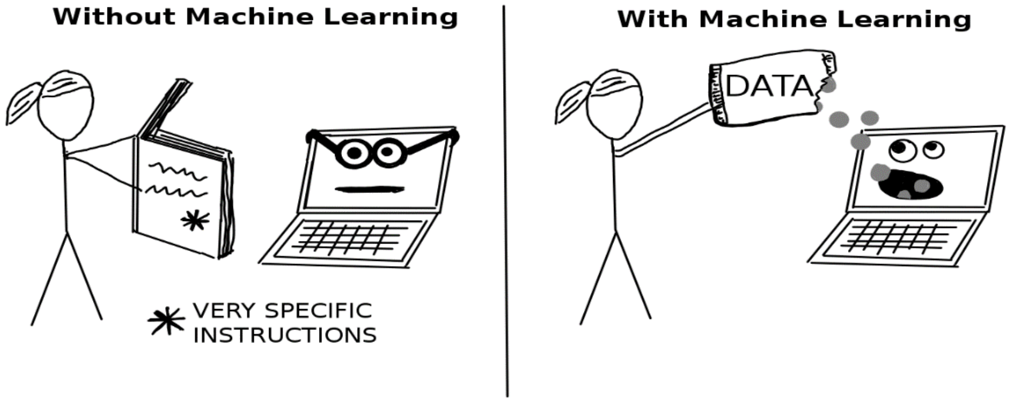
You need millions of numbers to describe a deep neural network, and there is no way to understand the model in its entirety. Other models, such as the **random forest**, consist of hundreds of decision trees that "vote" for predictions. To understand how the decision was made, you would have to investigate the votes and structures of each of the hundreds of trees. That just does not work no matter how clever you are or how good your working memory is. The best performing models are often blends of several models (**also called ensembles**) that cannot be interpreted, even if each single model could be interpreted. If you focus only on performance, you will automatically get more and more opaque models. The winning models on machine learning competitions are often ensembles of models or very complex models such as **boosted trees** or **deep neural networks.**

More concisely …

To avoid confusion due to ambiguity, here are some definitions of terms:

An **Algorithm** is a set of rules that a machine follows to achieve a particular goal. An algorithm can be considered as a recipe that defines the inputs, the output and all the steps needed to get from the inputs to the output. Cooking recipes are algorithms where the ingredients are the inputs, the cooked food is the output, and the preparation and cooking steps are the algorithm instructions.

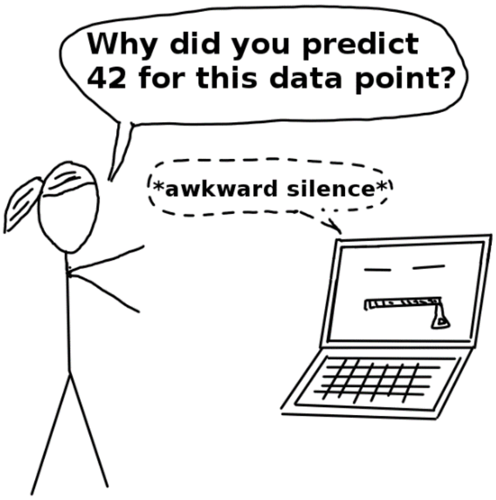
Machine Learning is a set of methods that allow computers to learn from data to make and improve predictions (for example cancer, weekly sales, credit default). Machine learning is a paradigm shift from "normal programming" where all instructions must be explicitly given to the computer to "indirect programming" that takes place through providing data.



A **Learner** or **Machine Learning Algorithm** is the program used to learn a machine learning model from data. Another name is "inducer" (e.g., "tree inducer").

A **Machine Learning Model** is the learned program that maps inputs to predictions. This can be a set of weights for a linear model or for a neural network. Other names for the rather unspecific word "model" are "predictor" or - depending on the task - "classifier" or "regression model". In formulas, the trained machine learning model is called $\hat{f}$ or $\hat{f}(x)$.

A **Black Box Model** is a system that does not reveal its internal mechanisms. In machine learning, "black box" describes models that cannot be understood by looking at their parameters (e.g., a neural network). The opposite of a black box is sometimes referred to as **White Box** or what can be also called as **interpretable** model**.**



Interpretable Machine Learning refers to methods and models that make the behavior and predictions of machine learning systems understandable to humans.

A **Dataset** is a table with the data from which the machine learns. The dataset contains the features and the target to predict. When used to induce a model, the dataset is called training data.

An **Instance** is a row in the dataset. Other names for 'instance' are: (data) point, example, observation. An instance consists of the feature values $x^{(i)}$ and, if known, the target outcome y_i.

The **Features** are the inputs used for prediction or classification. A feature is a column in the dataset. In most cases, features are assumed to be interpretable, meaning it is easy to understand what they mean, like the temperature on a given day or the height of a person. The interpretability of the features is a big assumption. But if it is hard to understand the input features, it is even harder to understand what the model does. The matrix with all features is called X and $x^{(i)}$ for a single instance.

The vector of a single feature for all instances is x_j and the value for the feature j and instance i is $x_j^{(i)}$.

The **Target** is the information the machine learns to predict. In mathematical formulas, the target is usually called $y\ or\ y_i$ for a single instance.

A **Machine Learning Task** is the combination of a dataset with features and a target. Depending on the type of the target, the task can be for example classification, regression, survival analysis, clustering, or outlier detection.

The **Prediction** is what the machine learning model "guesses" what the target value should be based on the given features. In this book, the model prediction is denoted by $\hat{f}\left(x^{(i)}\right) or\ \hat{y}$.

Cross-Validation is the process to select the optimal values of hyper-parameters is called model selection. if we reuse the same test dataset repeatedly during model selection, it will become part of our training data and thus the model will be more likely to over fit.

The overall data set is divided into:

1. the training data set
2. validation data set
3. test data set.

The training set is used to fit the different models, and the performance on the validation set is then used for the model selection. The advantage of keeping a test set that the model hasn't seen before during the training and model selection steps is that we avoid over-fitting the model and the model is able to better generalize to unseen data.