

Introduction to Machine Learning Top-Down Approach

A super smooth introduction to machine learning

[Maher](#)

You have 1 free story left this month.



Introduction to **Machine Learning**
Make your machine CREATIVE, With Maher

We couldn't have ever imagined how much computers will improve to become the monstrous machines we have today, they're not just getting better at what they do, they are also conquering other jobs.

In the time I wrote this article

- There are 4,156,513,325 Users on the internet, If you tried to count them it'd take more than 128 years.
- 1,755,606,975 websites on the internet.
- 168,673,726,872 Emails sent today only.
- 4,723,747,823 Google Searches today only.
- 4,033,234 Blog posts today only.

- 4,553,543,234 Videos views on youtube today only.

I can go on like this forever, and these numbers increase incredibly fast, A great amount of data exists that we can even make computers have insights about, we can program the computer to develop some kind of experience from going through this data and formulate a small simple **Brain** (dumb in most cases) but dedicated to solving a specific problem or even more than one as long as you're a ninja level programmer/researcher.

This **Brain** we often call it a model, it's most probably a **function**, if you have an enormous amount of data and want the brain/model to be complex and sophisticated, you can make it a much higher degree function (**aka neural network**) and by higher I'm not talking about the cubic or quadratic functions we took in school I mean much higher.

It 's not as complex as it sounds, everything can be as easy

The good thing is (**Introduction to Machine Learning**) is not the place to take you deep into this complex mathematical stuff.

First I would convince you that machine learning is a really interesting field and tell you some stuff about robots, face detections, SPAM mail detection and on and on but I want this to be short and to the point, so read the following, try to make a map in your mind and stay focused, starting now!

Define an outline

— the **MachineLearning-System** reads the **dataset** and optimizes the **Brain/Model** to solve the **problem**.

Focus on these keywords, we're going to discuss each one of them.

- **MachineLearning-System(ML-System)**: The Algorithm that controls the

whole process

- **Dataset:** The data provided to the system.
- **Model:** The brain or function we optimize to solve the problem.
- **Problem:** Surprisingly it's the problem we're trying to solve.

Machine Learning Systems

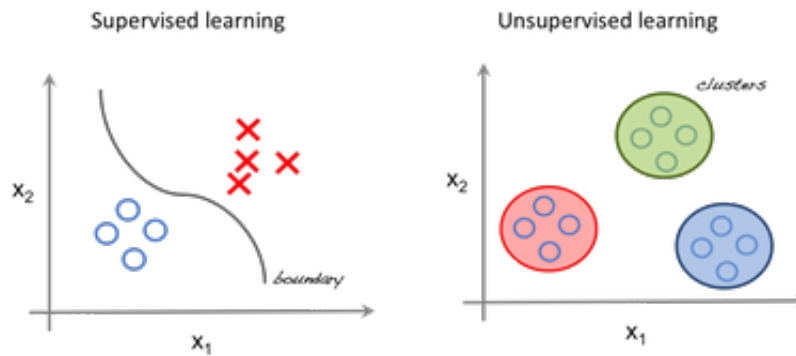
Let's start by talking about **Machine Learning-System (ML-System)**, It's simply an algorithm that does the job, iterates through the dataset, initialize the Model and feed the data to the model to learn from it.

Classifications of Machine Learning System:

we can classify Machine learning systems in more than one classification, as follows.

1. Whether or not the Model is trained with human supervision (**Supervised, UnSupervised, SemiSupervised** and **Reinforcement Learning**), with simpler words:

- In **Supervised** ML-System, The training data (dataset) you feed to the algorithm includes the desired solutions, called **labels**.
- In **UnSupervised** ML-System, The training data (dataset) you feed to the algorithm is **unlabeled** (the desired solution isn't included), and the system tries to learn without a teacher.
- In **SemiSupervised** ML-System, The training data (dataset) you feed to the algorithm is **partially labeled**.
- In **Reinforcement** ML-System, It's a little different, the model here is called *agent*, and its job is to perform actions and he gets rewards, The agent then learns as he's trying to maximize the rewards, it's like training your pet with treats.



2. Whether or not the Model can learn incrementally on the fly (**Online Learning, Batch Learning**), with simpler words:

- In **Batch Learning**, The Model is incapable of learning incrementally. First, we let the model train on all the data and then launch it to production.
- In **Online Learning**, The model is trained incrementally by feeding it instances sequentially, either individually or by small groups called *mini-batches*.

Dataset

It's the training data provided to the system to train on, We reduce the complexity of the dataset into **features** to be easy for the model to learn from the data

A feature is an individual measurable property or characteristic of a phenomenon being observed.

For example, let's use housing prices dataset, it's mainly a table where each row represents a house, columns = ['house_id', 'house_size', 'no_of_rooms', 'price'] and the **Problem** is to predict the price of a house.

Deciding which features to use is a really important step and you have to

consider a lot of things, I'll walk you through this step in another article, but we can start with features = [house_size, no_of_rooms] because it's a simple example.

The model we're using will try to have a sense of how these labels(outputs) can be estimated from the given features, in other words, how the price of the house is affected by its size and number of rooms in it. this process called, *Training*.

Problems we can face with Datasets

1. Insufficient Quantity of Training Data.

Models need thousands of examples to solve simple problems, and for more complex ones like image recognition they need millions of examples

In a famous paper published in 2001, Microsoft researchers Michele Banko and Eric Brill showed that very different machine learning algorithms including simple ones performed identically well on a complex problem once they were given enough data.

That shows how important is the Quantity of Training Data.

2. Nonrepresentative Training Data.

It is crucial to use training data that is representative of the cases you want to generalize to, The dataset should cover as many different cases as possible.

3. Poor-Quality Data.

which means it's full of unintended errors, noise, and outliers, this will make it harder for the model to detect patterns, and most probably will not perform well. Have some time to clean the data or discard the noisy parts.

4. Irrelevant features.

Features selection is a very important step, The model can detect

relations between the desired label(solution) and an irrelevant feature, that'll make the predictions more random.

5. **Overfitting.**

it means that the model performs well on the training data, but it does not generalize well. in other words, The model has studied the training data very well that it memorized it, This happens most probably because the model is too complex relative to the amount of data and its noisiness. (I'll tell you what to do if you encountered this problem in another article).

6. **Underfitting.**

It's obvious that it's the opposite of overfitting, this happens when The model is too simple to understand the data.

To know which model to choose for your project, **read this article** [Which machine learning model to use?](#)

Here are some models you can take a look at and I'll write articles to explain how they work.

Supervised learning algorithms examples

- k-Nearest Neighbors
- linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural Networks — NOTE: some neural networks can be unsupervised (e.g. autoencoders and restricted Boltzmann) or semisupervised.

Unsupervised learning algorithms examples

- Clustering (e.g. k-Means, Hierarchical Cluster Analysis HCA)
- Visualization and dimensionality reduction (e.g. Principal Component Analysis PCA)
- Association rule learning (e.g. Eclat)

You can read [5-Steps and 10-Steps, to Learn Machine Learning](#) article published in Towards Data Science Publication

Have a nice time Making your Machine ***CREATIVE***.