Only learning matters

What does learning exactly mean? Simply, we can say that learning is the ability to change according to external stimuli and remembering most of all previous experiences. So machine learning is an engineering approach that gives maximum importance to every technique that increases or improves the propensity for changing adaptively. A mechanical watch, for example, is an extraordinary artifact, but its structure obeys stationary laws and becomes useless if something external is changed. This ability is peculiar to animals and, in particular, to human beings; according to Darwin's theory, it's also a key success factor for the survival and evolution of all species. Machines, even if they don't evolve autonomously, seem to obey the same law.

Therefore, the main goal of machine learning is to study, engineer, and improve mathematical models which can be trained (once or continuously) with context-related data (provided by a generic environment), to infer the future and to make decisions without complete knowledge of all influencing elements (external factors). In other words, an agent (which is a software entity that receives information from an environment, picks the best action to reach a specific goal, and observes the results of it) adopts a statistical learning approach, trying to determine the right probability distributions and use them to compute the action (value or decision) that is most likely to be successful (with the least error).

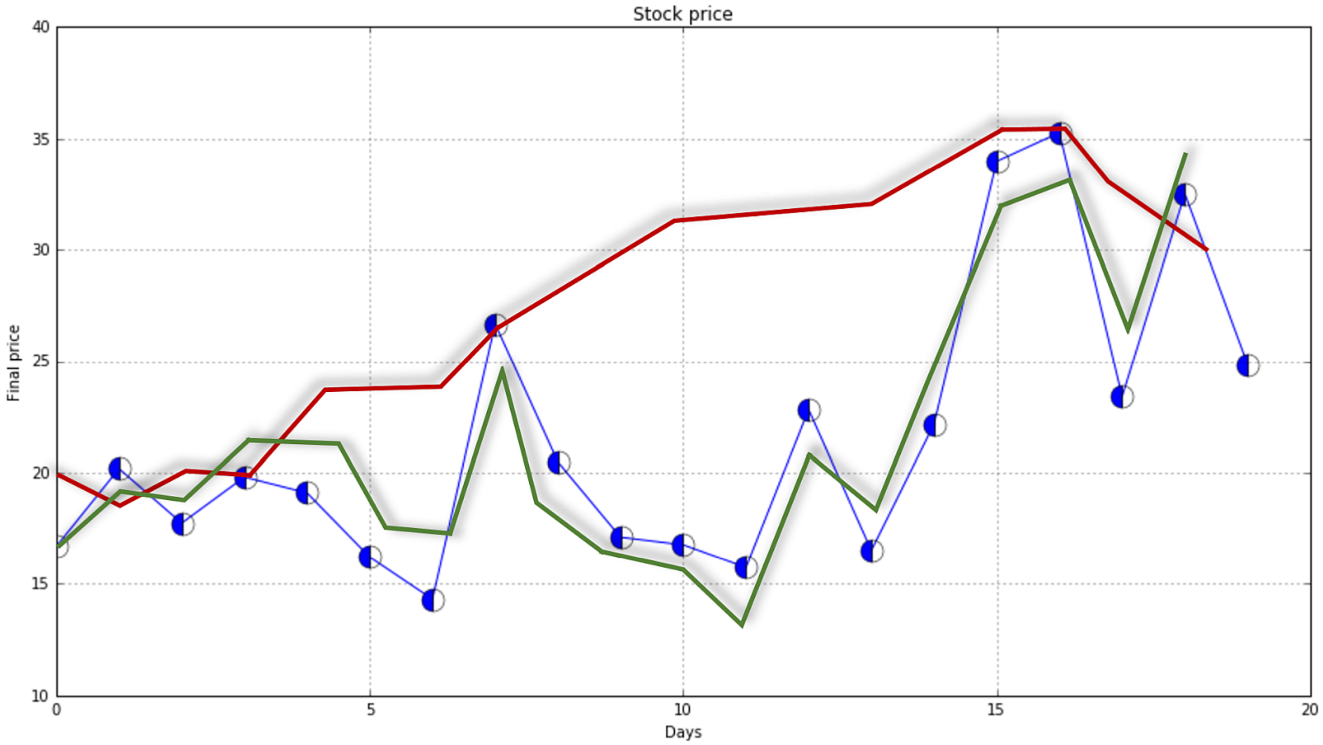
I do prefer using the term **inference** instead of **prediction** only to avoid the weird (but not so uncommon) idea that machine learning is a sort of modern magic. Moreover, it's possible to introduce a fundamental statement: an algorithm can extrapolate general laws and learn their structure with relatively high precision only if they affect the actual data. So the term **prediction** can be freely used, but with the same meaning adopted in physics or system theory. Even in the most complex scenarios, such as image classification with convolutional neural networks, every piece of information (geometry, color, peculiar features, contrast, and so on) is already present in the data and the model has to be flexible enough to extract and learn it permanently.

In the next sections, there's a brief description of some common approaches to machine learning. Mathematical models, algorithms, and practical examples will be discussed in later chapters.

Supervised learning

A supervised scenario is characterized by the concept of a teacher or supervisor, whose main task is to provide the agent with a precise measure of its error (directly comparable with output values). With actual algorithms, this function is provided by a training set made up of couples (input and expected output). Starting from this information, the agent can correct its parameters so as to reduce the magnitude of a global loss function. After each iteration, if the algorithm is flexible enough and data elements are coherent, the overall accuracy increases and the difference between the predicted and expected value becomes close to zero. Of course, in a supervised scenario, the goal is training a system that must also work with samples never seen before. So, it's necessary to allow the model to develop a generalization ability and avoid a common problem called **overfitting**, which causes an **overlearning** due to an excessive capacity (we're going to discuss this in more detail in the next chapters, however we can say that one of the main effects of such a problem is the ability to predict correctly only the samples used for training, while the error for the remaining ones is always very high).

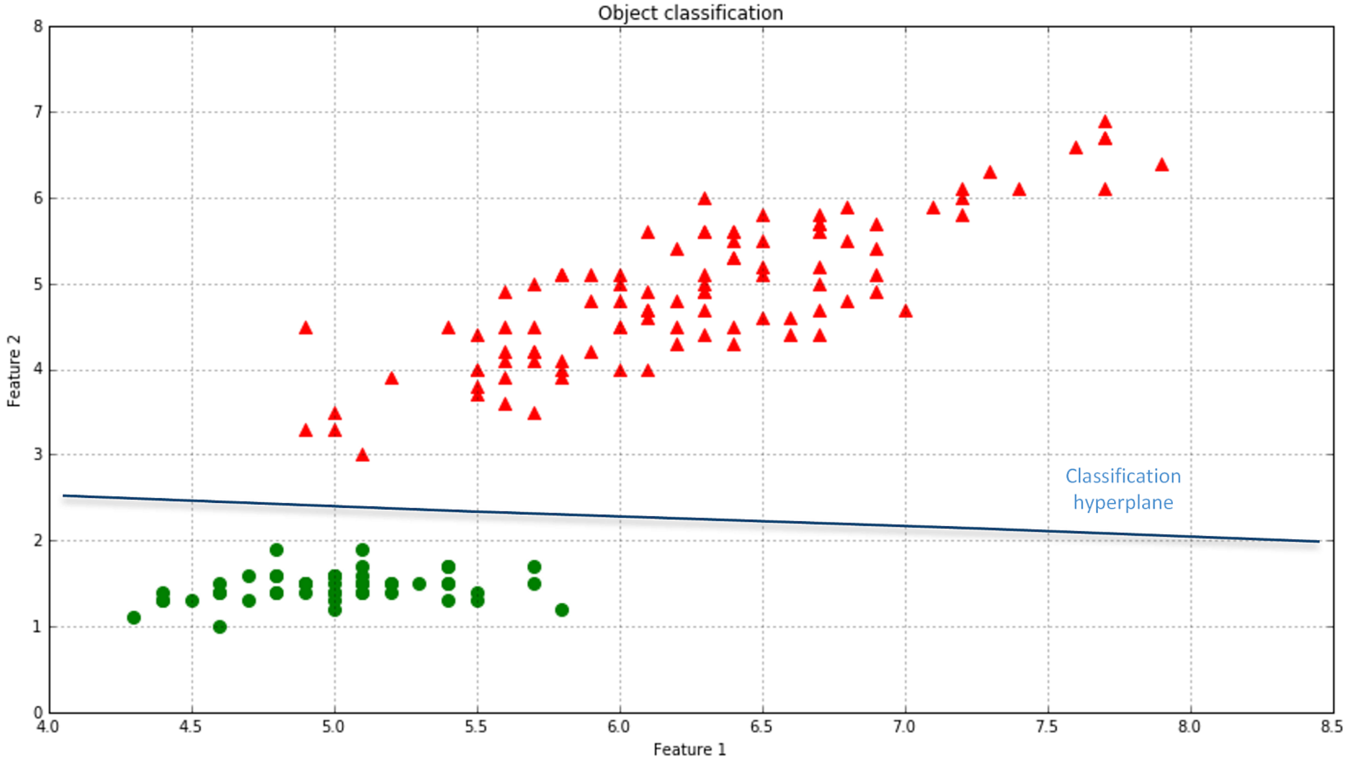
In the following figure, a few training points are marked with circles and the thin blue line represents a perfect generalization (in this case, the connection is a simple segment):



Two different models are trained with the same datasets (corresponding to the two larger lines). The former is unacceptable because it cannot generalize and capture the fastest dynamics (in terms of frequency), while the latter seems a very good compromise between the original trend and a residual ability to generalize correctly in a predictive analysis.

Formally, the previous example is called **regression** because it's based on continuous output values. Instead, if there is only a discrete number of possible outcomes (called **categories**), the process becomes a **classification.** Sometimes, instead of predicting the actual category, it's better to determine its probability distribution. For example, an algorithm can be trained to recognize a handwritten alphabetical letter, so its output is categorical (in English, there'll be 26 allowed symbols). On the other hand, even for human beings, such a process can lead to more than one probable outcome when the visual representation of a letter isn't clear enough to belong to a single category. That means that the actual output is better described by a discrete probability distribution (for example, with 26 continuous values normalized so that they always sum up to 1).

In the following figure, there's an example of classification of elements with two features. The majority of algorithms try to find the best separating hyperplane (in this case, it's a linear problem) by imposing different conditions. However, the goal is always the same: reducing the number of misclassifications and increasing the noise-robustness. For example, look at the triangular point that is closer to the plane (its coordinates are about [5.1 - 3.0]). If the magnitude of the second feature were affected by noise and so the value were quite smaller than 3.0, a slightly higher hyperplane could wrongly classify it. We're going to discuss some powerful techniques to solve these problems in later chapters.



Common supervised learning applications include:

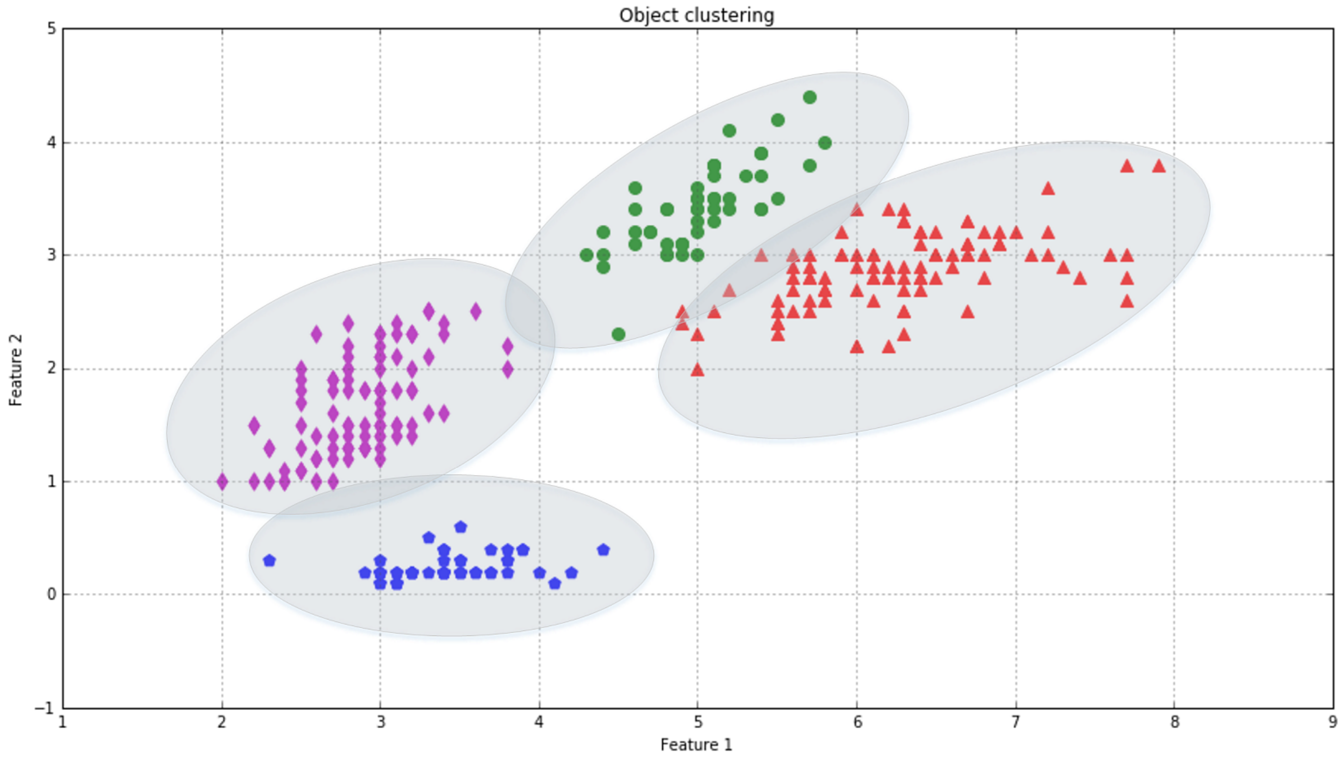
- Predictive analysis based on regression or categorical classification
- Spam detection
- Pattern detection
- Natural Language Processing
- Sentiment analysis
- Automatic image classification
- Automatic sequence processing (for example, music or speech)

Unsupervised learning

This approach is based on the absence of any supervisor and therefore of absolute error measures; it's useful when it's necessary to learn how a set of elements can be grouped (clustered) according to their similarity (or distance measure). For example, looking at the previous figure, a human being can immediately identify two sets without considering the colors or the shapes. In fact, the circular dots (as well as the triangular ones) determine a coherent set; it is separate from the other one much more than how its points are internally separated. Using a metaphor, an ideal scenario is a sea with a few islands that can be separated from each other considering only their mutual position and internal cohesion.

In the next figure, each ellipse represents a cluster and all the points inside its area can be labeled in the same way. There are also boundary points (such as the triangles overlapping the circle area) that need a specific criterion (normally a trade-off distance measure) to determine the corresponding cluster. Just as for classification with ambiguities (P and malformed R), a good clustering approach should consider the presence of outliers and treat them so as to increase both the internal coherence (visually, this means picking a subdivision that maximizes the local density) and the separation among clusters.

For example, it's possible to give priority to the distance between a single point and a centroid, or the average distance among points belonging to the same cluster and different ones. In this figure, all boundary triangles are close to each other, so the nearest neighbor is another triangle. However, in real-life problems, there are often boundary areas where there's a partial overlap, meaning that some points have a high degree of uncertainty due to their feature values.



Another interpretation can be expressed using probability distributions. If you look at the ellipses, they represent the area of multivariate Gaussians bound between a minimum and maximum variance. Considering the whole domain, a point (for example, a blue star) could potentially belong to all clusters, but the probability given by the first one (lower-left corner) is the highest, and so this determines the membership. Once the variance and mean (in other words, the shape) of all Gaussians become stable, each boundary point is automatically captured by a single Gaussian distribution (except in the case of equal probabilities). Technically, we say that such an approach maximizes the likelihood of a Gaussian mixture given a certain dataset. This is a very important statistical learning concept that spans many different applications, so it will be examined in more depth in the next chapter. Moreover, we're going to discuss some common clustering methodologies, considering both strong and weak points and comparing their performances for various test distributions.

Other important techniques involve the usage of both labeled and unlabeled data. This approach is therefore called semi-supervised and can be adopted when it's necessary to categorize a large amount of data with a few complete (labeled) examples or when there's the need to impose some constraints to a clustering algorithm (for example, assigning some elements to a specific cluster or excluding others).

Commons unsupervised applications include:

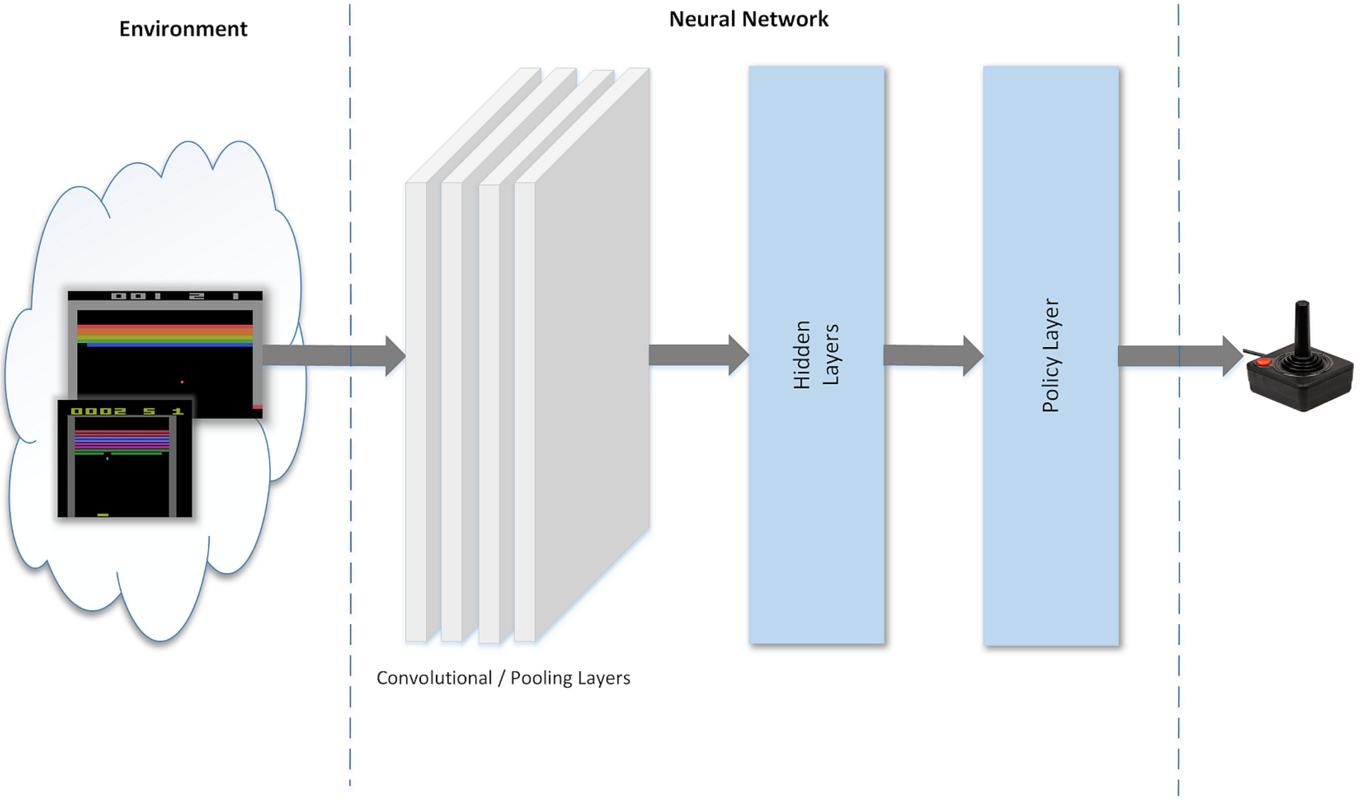
- Object segmentation (for example, users, products, movies, songs, and so on)
- Similarity detection
- Automatic labeling

Reinforcement learning

Even if there are no actual supervisors, reinforcement learning is also based on feedback provided by the environment. However, in this case, the information is more qualitative and doesn't help the agent in determining a precise measure of its error. In reinforcement learning, this feedback is usually called **reward** (sometimes, a negative one is defined as a penalty) and it's useful to understand whether a certain action performed in a state is positive or not. The sequence of most useful actions is a policy that the agent has to learn, so to be able to make always the best decision in terms of the highest immediate and cumulative reward. In other words, an action can also be imperfect, but in terms of a global policy it has to offer the highest total reward. This concept is based on the idea that a rational agent always pursues the objectives that can increase his/her wealth. The ability to **see** over a distant horizon is a distinction mark for advanced agents, while short-sighted ones are often unable to correctly evaluate the consequences of their immediate actions and so their strategies are always sub-optimal.

Reinforcement learning is particularly efficient when the environment is not completely deterministic, when it's often very dynamic, and when it's impossible to have a precise error measure. During the last few years, many classical algorithms have been applied to deep neural networks to learn the best policy for playing Atari video games and to teach an agent how to associate the right action with an input representing the state (usually a screenshot or a memory dump).

In the following figure, there's a schematic representation of a deep neural network trained to play a famous Atari game. As input, there are one or more subsequent screenshots (this can often be enough to capture the temporal dynamics as well). They are processed using different layers (discussed briefly later) to produce an output that represents the policy for a specific state transition. After applying this policy, the game produces a feedback (as a reward-penalty), and this result is used to refine the output until it becomes stable (so the states are correctly recognized and the suggested action is always the best one) and the total reward overcomes a predefined threshold.



We're going to discuss some examples of reinforcement learning in the chapter dedicated to introducing deep learning and TensorFlow.

❮ Previous Section (/book/big_data_and_business_intelligence/9781785889622/1/ch01lvl1s

Next Section ❯ (/book/big_data_and_business_intelligence/9781785889622/1/ch01lvl1sec1