Types of Machine Learning

The ways we train machines generally fall into three main methods: Supervised learning, semi-supervised learning, and unsupervised learning.

Supervised learning is when the training data has both the input and output. Returning to the image recognition example from above, the training set would include images with a goat and images without a goat, and each image would have features of a goat or not a goat. In this way you're supervising the machine's learning with both the input (e.g., explicitly telling the computer specific attributes of a goat: arch in back, size and position of head and ears, size and relative location of legs, presence and shape of backwardly arching horns, size and presence of a tail, and straightness and length of hair, etc.) and the output (labels of either goat or not goat) so the machine can learn to recognize the characteristics of a goat. In other words, we're getting the computer to learn a classification system we have created. *Classification algorithms* can sort email into folders (including a spam folder).

Most *regression algorithms* are so named for their continuous outputs that can vary within a range, such as temperature, length, blood pressure, body weight, profit, price, revenue, and so on. The above sales model example would rely on a regression algorithm because revenue – the outcome of interest, is a continuous variable.

Semi-supervised learning occurs when the training data input is incomplete, meaning some portion of it does not have any labels. This can be an important check on bias in data when humans are labeling it since the algorithms will also learn from unlabeled data. Gargantuan tasks like classifying webpages, genetic sequencing, and speech recognition are all tasks that can benefit greatly with a semi-supervised approach to learning.

Unsupervised learning happens when the training data input contains no output labels at all (most "big data" often lacks output labels), which means you want the machine to learn how to do something without telling it specifically how to do it. This is useful for identifying patterns and structures in the data that might otherwise escape human detection. Identified groupings, clusters and categories of data points can result in "ah-hah" insights. In the goat example, the algorithm would evaluate different animal groups and label goats and other animals on its own. Unlike supervised learning, there is no mention of the specific characteristics of a goat. This is an example of *feature learning* that automatically discovers the representations needed for feature detection or classification from raw data, thereby allowing the machine to both learn the features and then use them to perform a specific task.

Another example is *dimensionality reduction* that lessens the number of random variables under consideration in a set of data by identifying a group of principal variables. For reasons beyond the scope of this paper, sparsity is always an objective in model building for both parametric and non-parametric models. Unsupervised machine learning algorithms have made their way into business problem-solving and have proven especially useful in digital marketing, ad tech, and exploring customer information. Software platforms and apps that make use of unsupervised include Lotame (real-time data management) and Salesforce (customer relationship management software).

Those are three rather broad ways of classifying different kinds of machine learning. There are other ways to provide a taxonomy of machine learning types, but there is no widespread agreement on the best way to classify them. For this reason, there are lots of other kinds of machine learning, a small sampling of which are briefly described below:

- ***Reinforcement learning:*** The algorithm receives positive and negative feedback from a dynamic environment as it takes actions. The observations provide guidance to the algorithm. This is the kind of learning that is vital to autonomous cars or to machines playing games against a human opponent. It is also what most people probably have in mind when they think of AI. “Reinforced ML uses the technique called exploration/exploitation. The mechanics are simple – the action takes place, the consequences are observed, and the next action considers the results and errors of the first action.”
- ***Active learning:*** When it is impractical or too expensive for a human to label the training data, the algorithm can be set up to query the “teacher” when it needs a data label. This makes active learning a variation of supervised learning.
- ***Meta learning:*** The algorithm learns to learn. It can learn its own inductive bias based on previous experience by evaluating different learning paths, including its own.
- ***Transduction:*** This variation on supervised learning attempts to predict new outputs based on the training inputs/outputs as well as new inputs. Unlike inductive models that go from observed training cases to general rules, a transduction goes from specific observed training cases to specific test cases.

- ***Deep learning:*** When the machine can take what it has learned and apply it to data it has not "seen" before (i.e., when the algorithm is applied), usually through the incorporation of Convolutional Neural Networks (CNNs), Deep Belief Networks (DBN), Deep Boltzman Machine (DBM), or Stacked Auto Coders.
- ***Robot learning:*** Algorithms that create their own "curricula" or sequences of learning experiences to acquire new skills through self-guided exploration and/or interacting with humans.
- ***Ensemble learning:*** This approach combines multiple models to improve predictive capabilities compared to using a single model. "Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to *decrease variance* (bagging), *decrease bias* (boosting), or *improve predictions* (stacking)."

Today's powerful computer processing and colossal data storage capabilities enable ML algorithms to charge through data using brute force. At a micro-level, the evaluation typically consists of answering a yes or no question. Most ML algorithms employ a *gradient descent* approach known as *back propagation*. To do this efficiently, a sequential iteration process is followed.

An analogy may help the reader better understand this class of algorithms in lieu of multivariate differential calculus: a car traverses through a figurative mountain range with peaks and valleys (loss function). The algorithm helps the car navigate through different locations (parameters) to find the lowest point in the mountain range by finding the path with the steepest descent. To overcome the problem of the car getting stuck in a valley (local minimum) that is not the lowest valley (global minimum), the algorithm sends multiple cars to different locations, each having the same mission. After every turn, the algorithm learns whether the cars' pathway helps or hinders getting to the lowest point. More technically, the goal of the gradient descent algorithm is to minimize the *loss function*. To improve the accuracy of the prediction, the parameters are *hyper-tuned* where different model families are used to find a more efficient path to the global minimum. Sticking with the mountain range analogy, an alternative route may fit the mountain topology better than others, thereby enabling a car to find the lowest point faster. Most of these ML models make relatively few assumptions about the data (e.g., continuity and differentiability are assumed for gradient descent algorithms) or virtually no assumptions (e.g., most clustering algorithms, such as K-means or random forests); the essential characteristic of what makes it "machine learning" is the consecutive processing of data using trial and error. Stochastic (can be analyzed statistically) models make assumptions about how the data is

distributed: “I’ve seen that mountain range before, and I don’t need to send a whole fleet of cars to accomplish the mission.”
It should make sense to the reader how iteration may lead to more precise predictions than stochastic methods: the exact routes are being calculated versus estimated with statistics. However, because the loss (aka cost) function will usually be applied to a slightly different mountain range, the statistical models, which are more generalizable, may ultimately lead to a superior prediction (or interpolation). However, while there is a clear difference between statistics and non-parametric ML, the iterative elements of non-parametric ML algorithms may be used in a statistical model and, conversely, non-parametric ML algorithms may contain stochastic elements.

A fuller understanding of the differences between these machine learning variations and their highly specific, continuously evolving algorithms requires a level of technical knowledge beyond the scope of this paper. A more fruitful avenue of inquiry here is to explore the relationship of ML and statistics as they relate to AI.