TYPES OF MACHINE LEARNING:

* Based on human supervision(supervised, unsupervised, reinforcement learning)
* They can learn incrementally on fly(online versus batch learning)
* Compare new data points with known data points, detect patterns in training data, and build a predictive model(instance-based versus model-based learning)

SUPERVISED/UNSUPERVISED LEARNING:

This is a classification according to the amount and type of supervision they get during training, based on which they are classified as supervised, unsupervised, semi-supervised, and reinforcement learning.

1. SUPERVISED LEARNING:

The data we feed to the algorithm has the desired solution called labels.

A typical supervised task is classification. The spam filter has two classes of spam or ham where the algorithm is trained based on this example and it must learn how to classify the new email, whether it is spam or not.

Another typical task is regression where the algorithm must predict a target numeric value such as the price of a car given a set of features called predictors.

In machine learning an attribute is a data type (e.g.,” mileage”) while a feature has several meanings depending on the context, but generally means attribute plus its value(e.g., mileage”=1500).

Some regression algorithms can also be used for classification and vice versa.

The most important supervised learning algorithms are:

* K-nearest neighbors
* Linear regression
* Logistic regression
* Support vector machine(SVMs)
* Decision Trees and Random Forests
* Neural Networks

1. UNSUPERVISED LEARNING:

The training data is unlabeled. The system tries to learn without a teacher.

The most important unsupervised learning algorithms are:

* Clustering
* K-means
* Hierarchical clustering analysis (HCA)
* Expectation maximization
* Visualization and dimensionality reduction
* Principal component analysis (PCA)
* Kernel PCA
* Locally-linear embedding (LLE)
* t-distributed stochastic neighbor embedding (t-SNE)
* Association rule learning
* Apriori
* Eclat

For example, say you have a lot of data about your blog's visitors. You may want to run a clustering algorithm to try to detect groups of similar visitors. At no point do you tell the algorithm which groups a visitor belongs to, the algorithm finds those connections without your help. For example, 40% of your visitors are males who love comics and generally read your book in the evening while 20% of your visitors are young sci-fi lovers who visit during the weekends.

The hierarchical clustering algorithm may also subdivide each group into smaller groups which may help you target your posts for each group.

Visualization algorithms are also good examples of unsupervised learning algorithms: you feed complex and unlabeled data and they output a 2D or 3d representation of your data that can be easily plotted.

Dimensionality reduction in which the goal is to simplify the data without losing too much information. One way to do this is to merge several correlated features into one.

For example, a car's mileage may be very correlated with its age so the dimensionality reduction algorithm will merge them into one feature that represents the car's wear and tear. This is called feature extraction.

Another important unsupervised task is anomaly detection.

For example, detecting usual credit card transactions to prevent fraud, catching manufacturing defects, or automatically removing outliers from a dataset before feeding it to another learning algorithm. The system is trained with a normal instance, when it sees a new instance it can tell whether it looks like a normal one or it is likely an anomaly.

Another common unsupervised task is association rule learning, in which the goal is to dig into large amounts of data and discover interesting relations between attributes.

For example, suppose you own a supermarket. Running an association rule on your sales logs may reveal that people who purchase barbecue sauce and potato chips also tend to buy steak. Thus you may know to place these items close to each other.

3. SEMISUPERVISED LEARNING :

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called semi-supervised learning.

For example, Google Photos where automatically recognizes that person A is in photos 1, and 2,6, while person B shows up in photos 3,5,8. This is the unsupervised part of the algorithm (clustering) Now the system needs is for you to tell it who these people are. Just label per person, and it can name everyone in every photo, which is useful for searching photos.

1. REINFORCEMENT LEARNING:

In reinforcement learning, a system called an agent observes the environment, selects and performs actions, and receives rewards and negative rewards in return. It must learn the best strategy called policy to get the most reward over time. A policy defines what action the agent should choose in a given situation.

For example, robots implement reinforcement learning algorithms to learn how to walk. DeepMind's AlphaGo program is a good one that won against the champion. It learned its winning policy by analyzing millions of games and then playing many games against itself. The learning was turned off during the games against the champion.

BATCH AND ONLINE LEARNING:

BATCH LEARNING:

In batch learning, the system is incapable of learning incrementally: it must be trained using all the available data. A **batch** refers to a set of data used in one training session. In batch learning, the "batch" is typically the entire dataset. This will generally take a lot of time and computing resources, so it is typically done offline. First, the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned. This is called offline learning.

If you want a batch learning system to know about new data (such as a new type of spam), you need to train a new version of the system from scratch on the full dataset (not just the new data, but also the old data), then stop the old system and replace it with the new one.

ONLINE LEARNING:

In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or in small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives. Online learning is great for systems that receive data as a continuous flow (e.g., stock prices) and need to adapt to change rapidly or autonomously. It is also a good option if you have limited computing resources: once an online learning system has learned about new data instances, it does not need them anymore, so you can discard them.

One important parameter of online learning systems is how fast they should adapt to changing data: this is called the learning rate. If you set a high learning rate, then your system will rapidly adapt to new data, but it will also tend to quickly forget the old data (you don’t want a spam filter to flag only the latest kinds of spam ).

Conversely, if you set a low learning rate, the system will have more inertia; that is, it will learn more slowly, but it will also be less sensitive to noise in the new data or to sequences of outliers.

INSTANCE-BASED VERSUS MODEL-BASED LEARNING:

INSTANCE-BASE LEARNING:

Instance-based learning is like solving problems by remembering examples you've seen before. Instead of learning a general rule or formula, the algorithm keeps the data it has been shown and uses it directly to make predictions. It stores the data and compares new things to what it already knows.

For example, Imagine you want to decide if a fruit is an apple or a pear. You have a list of fruits you've seen before, like:

* Apple: red, round
* Pear: green, oval

When you see a new fruit, you compare it to the examples you already have and choose the one it looks most like.

MODEL-BASED LEARNING:

Another way to generalize from a set of examples is to build a model of these examples, and then use that model to make predictions. This is called model-based learning.