Topic - 02
Introduction to Machine Learning: Concept & Fundamentals -- Classification & Terminologies

+88 01831-661534

jehadfeni@gmail.com

Attributes vs Features

In Machine Learning an ***ATTRIBUTE*** is a data type (e.g., “Price”), while a ***FEATURE*** has several meanings depending on the context, but generally means an attribute plus its value (e.g., “Price = 85,50,000”).

Many people use the words attribute and feature to represent the same thing.

Dimensionality Reduction

In many cases, we have to reduce our data without losing information in order to make the training faster. Easiest way of it is merging several related data as one.

For example, a House Price is connected with the Age of the House. So, we merge those 2 together and get a new House-Price feature. This process is called **FEATURE EXTRACTION**.

We will always try to achieve Dimensionality Reduction before starting any Machine Learning process (For example, any supervised process).

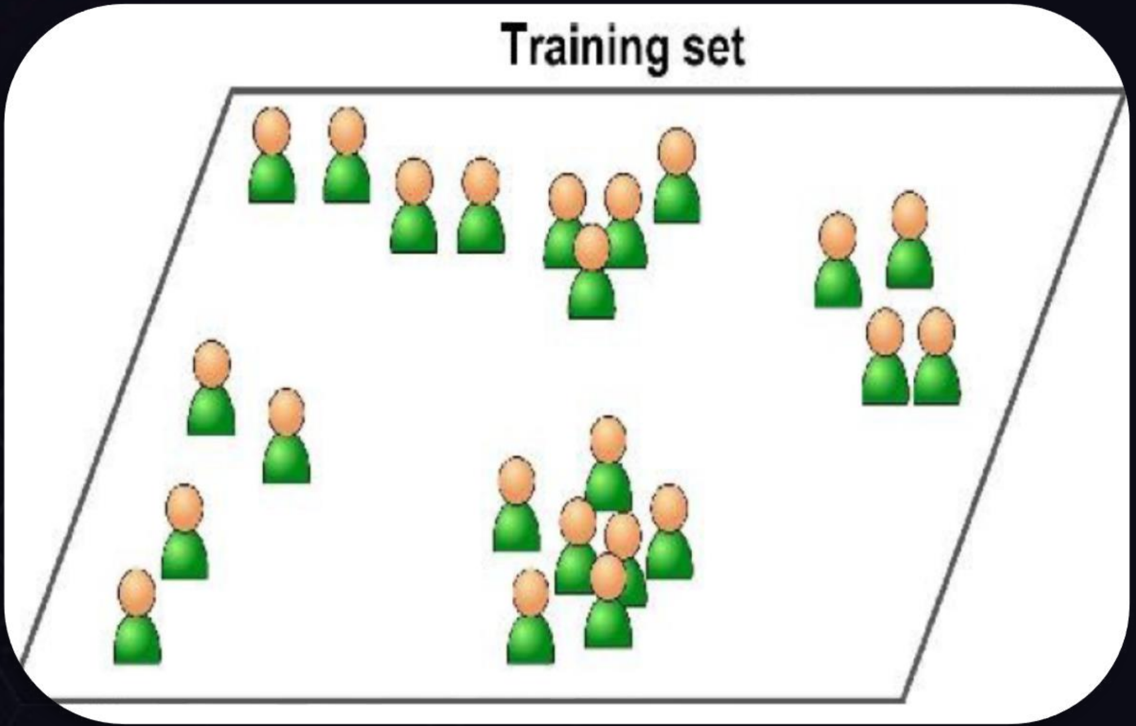
It will run much faster, the data will take up less disk and memory space, and in some cases it may also perform better.

Unsupervised Learnings - Examples

Let's say you want to categorize all of your Blog visitors into different groups of similar attributes. This process is called ***Clustering***. We don't give any input to the algorithm.

It tries to find the connection among the data and give output.

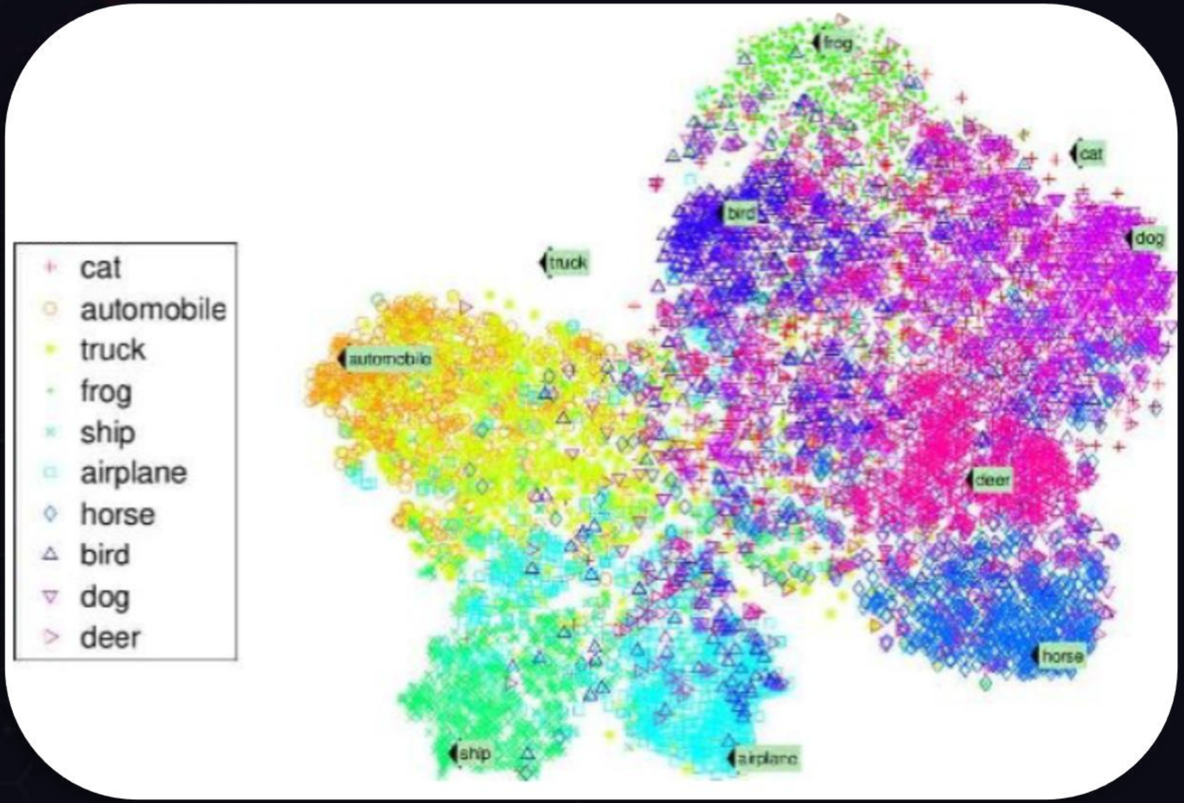
For example: You may find out that 30% of your blog visitors are female who love TRAVEL blogs in night, whilst 80% of your blog visitors are male who love SPORTS blog.



Unsupervised Learnings - Examples Cont.

A good example of Unsupervised Learning is Visualization Algorithm. You feed them UNLABELLED complex data and they will generate pattern which will help you to distinguish them. That way, both 2D and 3D representation can be obtained which can easily be plotted.

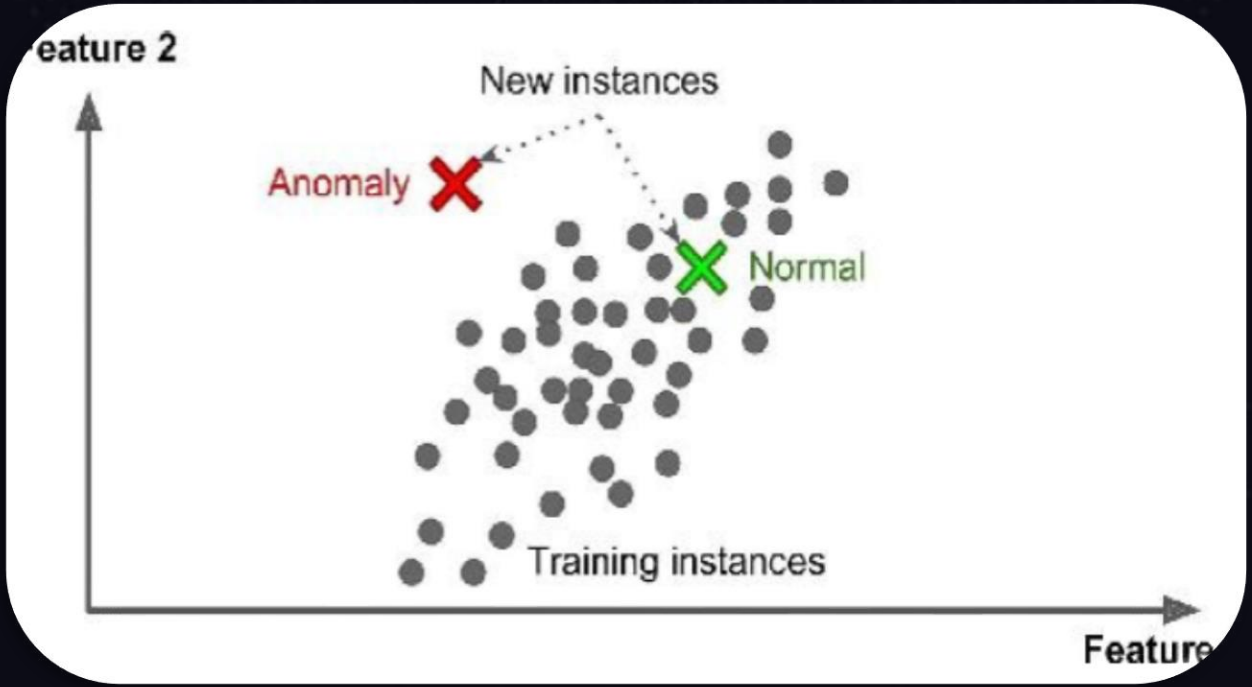
These algorithms keep as much structure
as possible so you can understand how the data is organized and identify unexpected patterns



Unsupervised Learnings - Examples Cont..

Anomaly Detection is another good task of unsupervised learning.

For example: Finding credit card fraud or catching manufacturing defects. First, the system is trained with all the normal instances. When a new instance comes, it can tell whether it is a normal instance or an anomaly.



Association Rule Learning

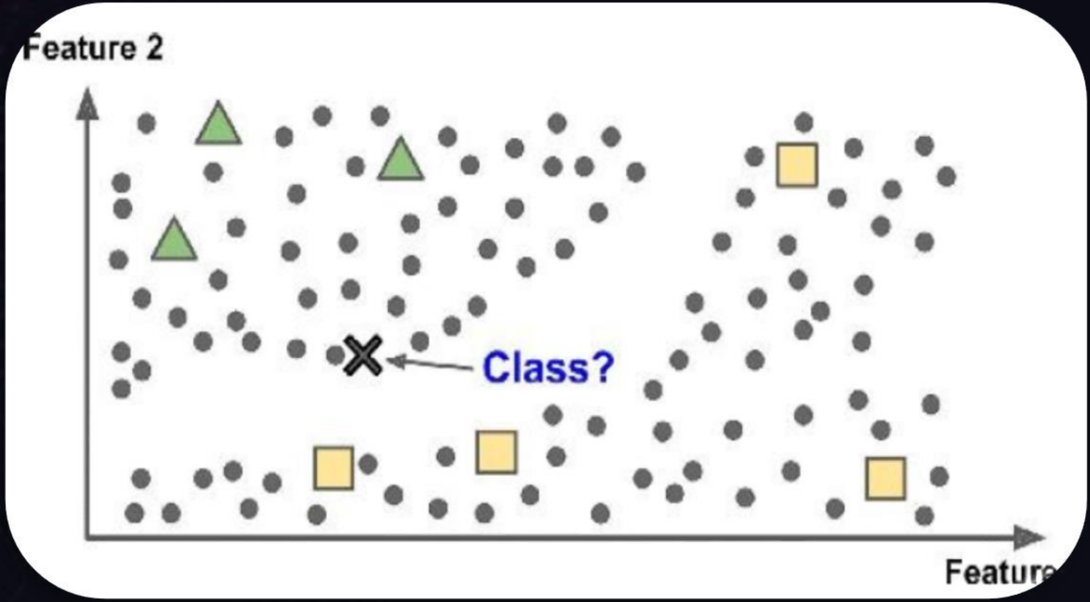
A common unsupervised task is ***Association Rule Learning***. There we go thoroughly into large amount of data and find relationship between ATTRIBUTES.

For example, you own a Super Shop. Running an Association Rule Learning into your Sales Log may help you find out that people who buy Barbecue Sauce and Potato Chips also buy Steak most of the time.

So, re-organizing your Shop and keeping these items together will help you increase your sales many times.

Semi-supervised Learnings

There are some algorithms out there which can deal with training data which are Partially labelled (only little of the data is labeled, rest are unlabeled). This process is called Semi-supervised learning.



For example, in Google photos, if you upload a lot of photos, it will first identify how many people are available in all the photos and giving their Thumbnail to identify. This is an unsupervised process.

Now, if you just TAG with NAME, who is who, then it can identify all the people in all the photos. Now, you can easily search people with their name.

Deep Belief Networks (DBNs)

Most Semi-supervised learning algorithms are combinations of Unsupervised Algorithms and Supervised Algorithms.

For example, Deep Belief Networks (DBNs) are based on unsupervised components called Restricted Boltzmann Machines (RBMs) stacked on top of one another.

RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

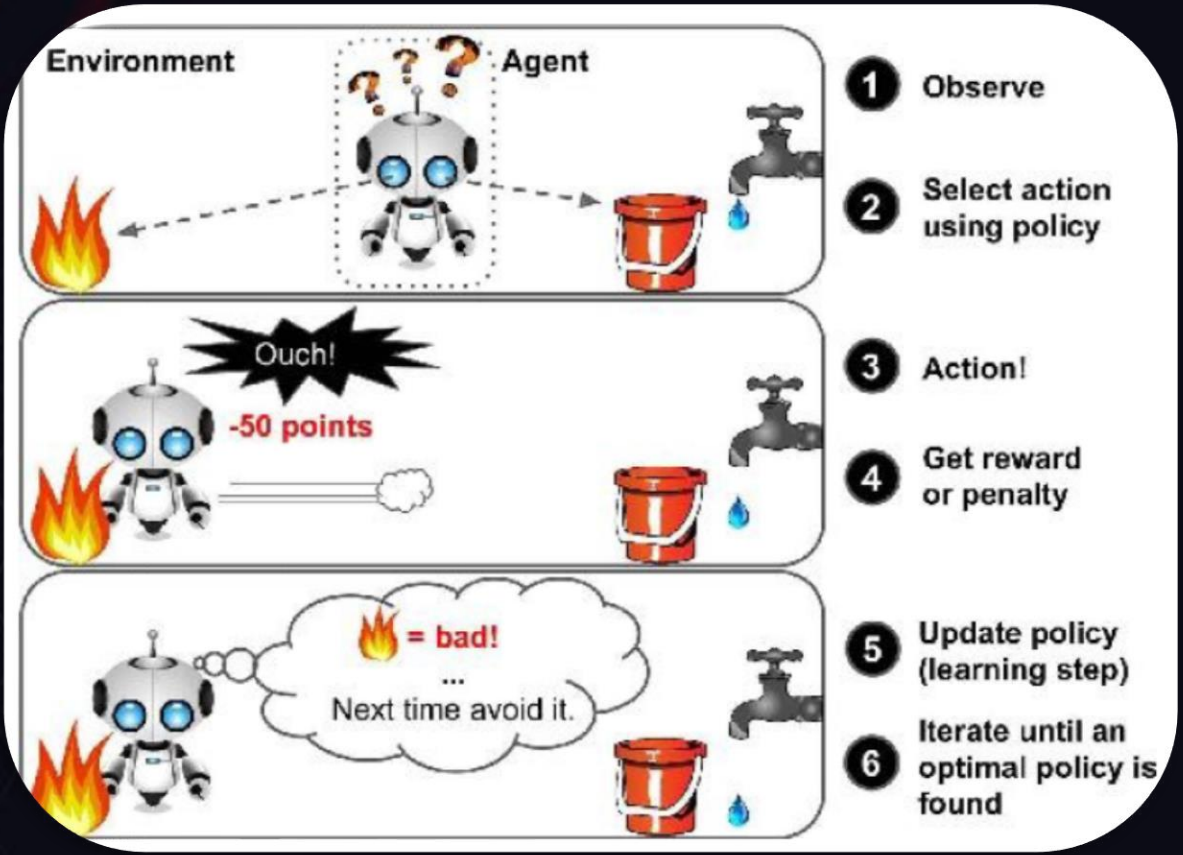
Reinforcement Learning

This is a totally different kind of Machine Learning process. In this case, an ***AGENT*** is appointed.

AGENT can –

- Observe the environment
- Select and perform actions
- Get rewards in return (or penalties in the form of negative rewards).

It must then learn by itself what is the best strategy, called a ***POLICY***, to get the most reward over time. A POLICY defines what action the agent should choose when it is in a given situation.



Reinforce Learning – Example

Many robots implement Reinforcement Learning algorithms to learn how to walk.

DeepMind's AlphaGo program is also a good example of Reinforcement Learning. It made the headlines in March 2016 when it beat the world champion Lee Sedol at the game of Go. ***GO*** is considered the hardest mind games in the whole world with Infinite possibilities.

It learned its winning policy by –
Analyzing millions of games, and then playing many games Against Itself.

Note that learning was turned off during the games against the champion; AlphaGo was just applying the policy it had learned.

Classification – based on Incremental Learning

Based on whether or not the system can learn ***INCREMENTALLY*** from a stream of incoming data, Machine Learning is divided into:

Batch Learning

Online Learning

Batch Learning

In batch learning, the system is incapable of learning incrementally. It must be trained using all the available data. 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.

Advantages & Disadvantages of Batch Learning

Advantages:

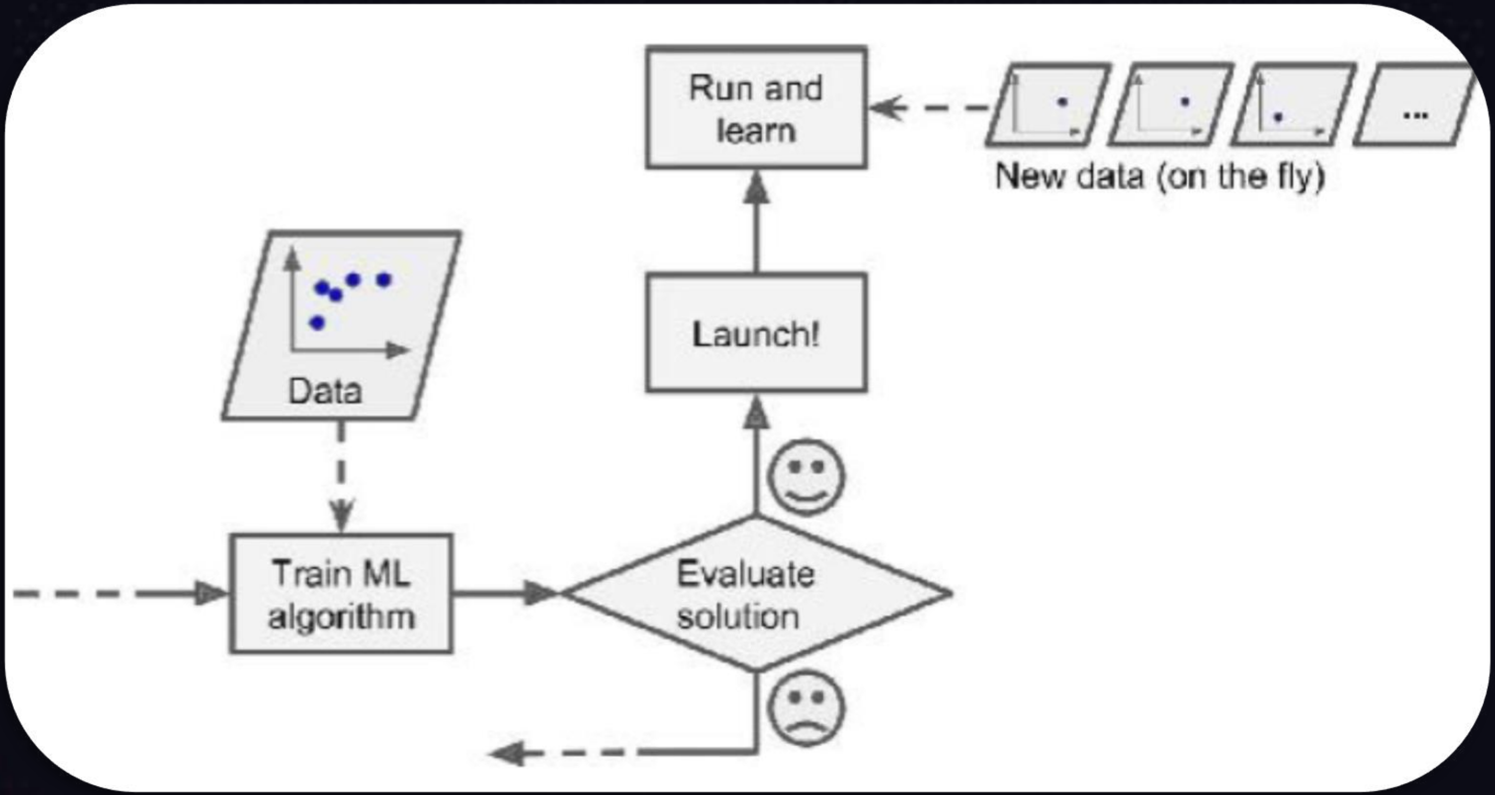
- If your data doesn’t have continuous update, then Batch Learning is the best solution
- Easy to maintain and the old model can be replaced with the new one very easily (with just replacing the old file).
- Works good for a generic problem. For example: Speech Recognition

Disadvantages:

- This solution is simple and often works fine, but training using the full set of data can take many hours. Doesn’t work well if your data needs continuous update (24 hours). For example: Stock Price Prediction.
- Training on the full set of data requires a lot of computing resources (CPU, memory space, disk space, disk I/O, network I/O)
- If the amount of data is huge, it may even be impossible to use a batch learning algorithm (If the training takes several days to finish)
- If your system needs to be able to learn autonomously and it has limited resources (e.g., a smartphone app), then carrying around large amounts of training data and taking up a lot of resources to train for hours every day is quite impossible.

Online Learning

In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or by 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 Cont.

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 (unless you want to be able to roll back to a previous state and "replay" the data).

This can save a huge amount of space.

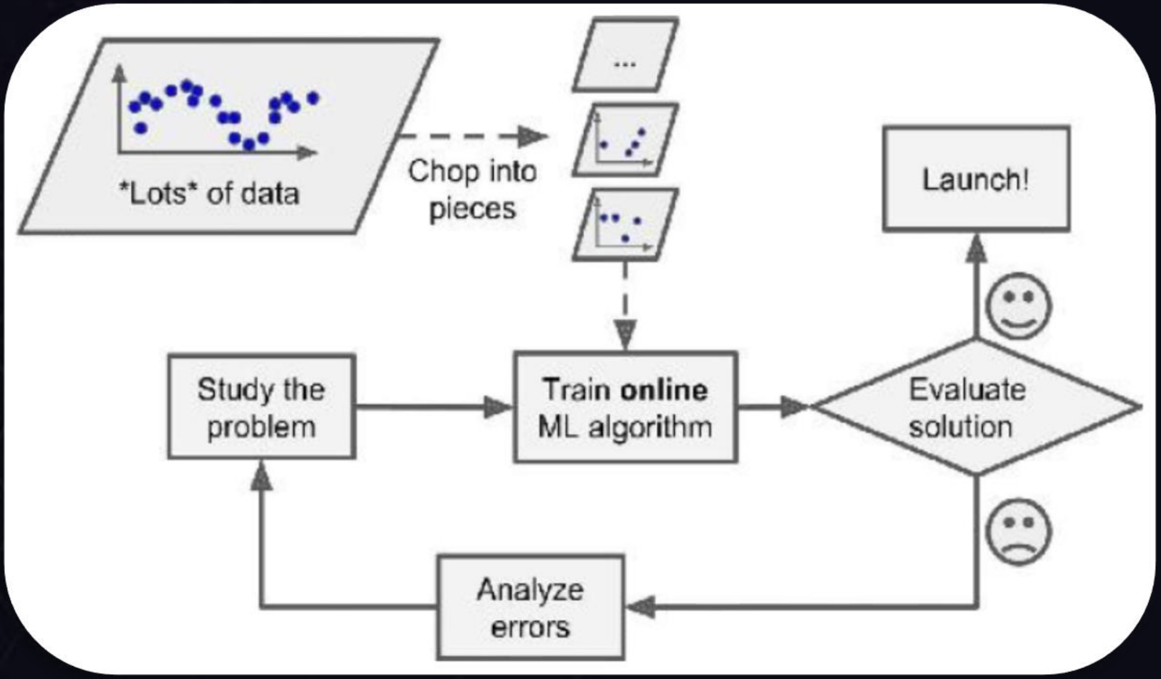
Online learning algorithms can also be used to train systems on huge datasets that cannot fit in one machine's main memory (this is called ***OUT-OF-CORE*** learning). The algorithm loads part of the data, runs a training step on that data, and repeats the process until it has run on all of the data.

Online Learning Cont..

One important parameter of online learning is how fast they can 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 it was shown).

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 nonrepresentative data.



Online Learning is Actually OFFLINE

Even though the name of the Process is ONLINE LEARNING, the process is usually done ***OFFLINE*** (i.e., not on the live system).

So online learning can be a confusing name.

We can alternatively call it ***INCREMENTAL LEARNING.***

Advantages & Disadvantages of Online Learning

Advantages:

- Easy for maintaining solutions that need continuous update
- Don’t need a lot of computing power and can learn from the New Data
- Best for Live Solutions
- Works easily on low resource devices like Smartphones or rover on the Mars which don’t have a lot of processing power

Disadvantages:

- If bad data is fed to the system, the system’s performance will gradually decline.
- If we are talking about a live system, clients will notice the downfall of the system that occurs due to the bad data.
  For example, bad data could come from a malfunctioning sensor on a robot, or from someone spamming a search engine to try to rank high in search results. To reduce this risk, you need to monitor your system closely and promptly switch learning off (and possibly revert to a previously working state) if you detect a drop in performance.
- You may also have to monitor the input data and react to abnormal data (e.g., using an anomaly detection algorithm).