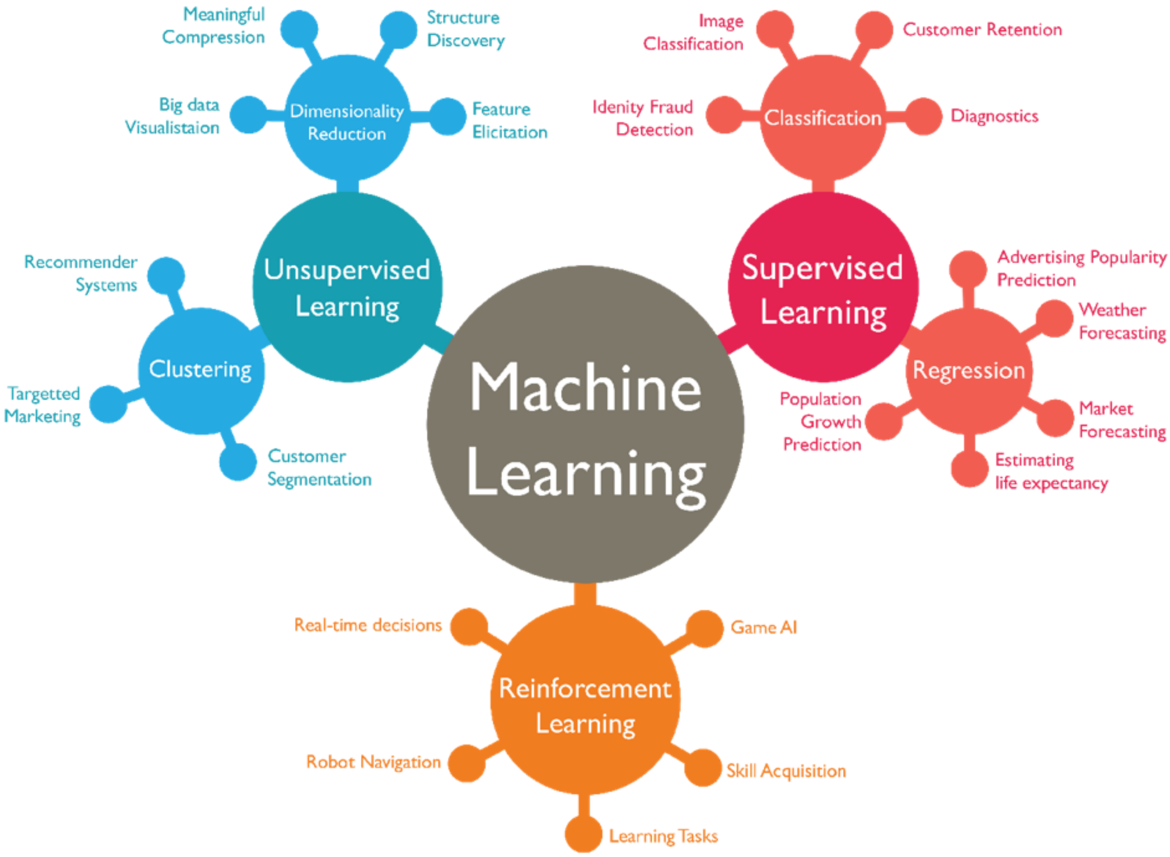
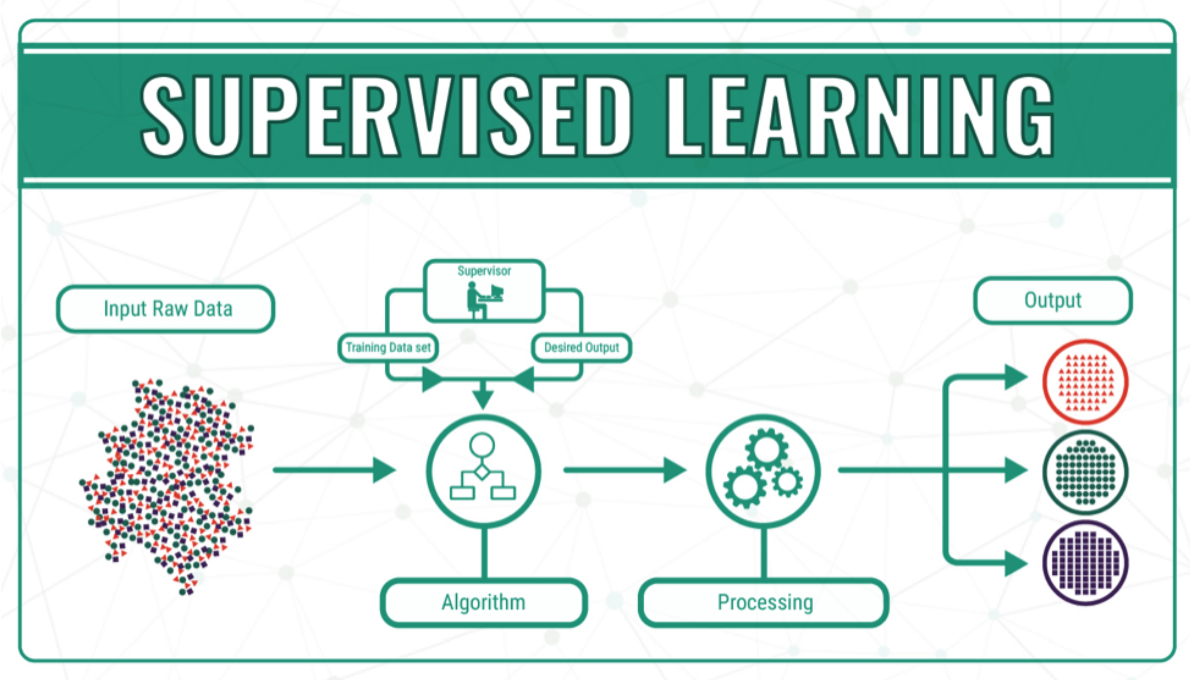
Types of Machine Learning

At a high-level, machine learning is simply the study of teaching a computer program or algorithm how to progressively improve upon a set task that it is given. On the research-side of things, machine learning can be viewed through the lens of theoretical and mathematical modeling of how this process works. However, more practically it is the study of how to build applications that exhibit this iterative improvement. There are many ways to frame this idea, but largely there are three major recognized categories: supervised learning, unsupervised learning, and reinforcement learning.



Supervised Learning



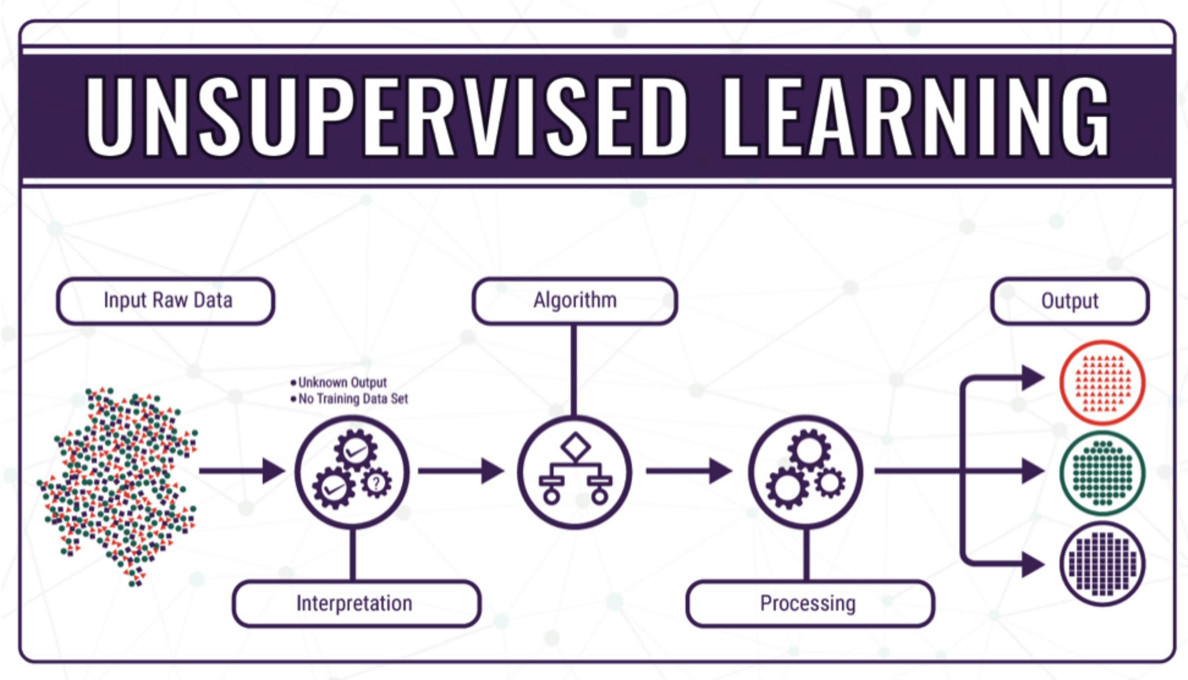
Supervised learning is the most popular paradigm for machine learning. It is the easiest to understand and the simplest to implement. It is very similar to teaching a child with the use of flash cards.

Given data in the form of examples with labels, we can feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels. When fully-trained, the supervised learning algorithm will be able to observe a new, never-before-seen example and predict a good label for it.

Supervised learning is often described as task-oriented because of this. It is highly focused on a singular task, feeding more and more examples to the algorithm until it can accurately perform on that task. This is the learning type that you will most likely encounter, as it is exhibited in many of the following common applications:

- **Advertisement Popularity:** Selecting advertisements that will perform well is often a supervised learning task. Many of the ads you see as you browse the internet are placed there because a learning algorithm said that they were of reasonable popularity (and clickability). Furthermore, its placement associated on a certain site or with a certain query (if you find yourself using a search engine) is largely due to a learned algorithm saying that the matching between ad and placement will be effective.
- **Spam Classification:** If you use a modern email system, chances are you've encountered a spam filter. That spam filter is a supervised learning system. Fed email examples and labels (spam/not spam), these systems learn how to preemptively filter out malicious emails so that their user is not harassed by them. Many of these also behave in such a way that a user can provide new labels to the system and it can learn user preference.
- **Face Recognition:** If you use Facebook or some other social media platform, most likely your face has been used in a supervised learning algorithm that is trained to recognize your face. Having a system that takes a photo, finds faces, and guesses who that is in the photo (suggesting a tag) is a supervised process. It has multiple layers to it, finding faces and then identifying them, but is still supervised nonetheless.

Unsupervised Learning



Unsupervised learning is very much the opposite of supervised learning. It features no labels. Instead, our algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.

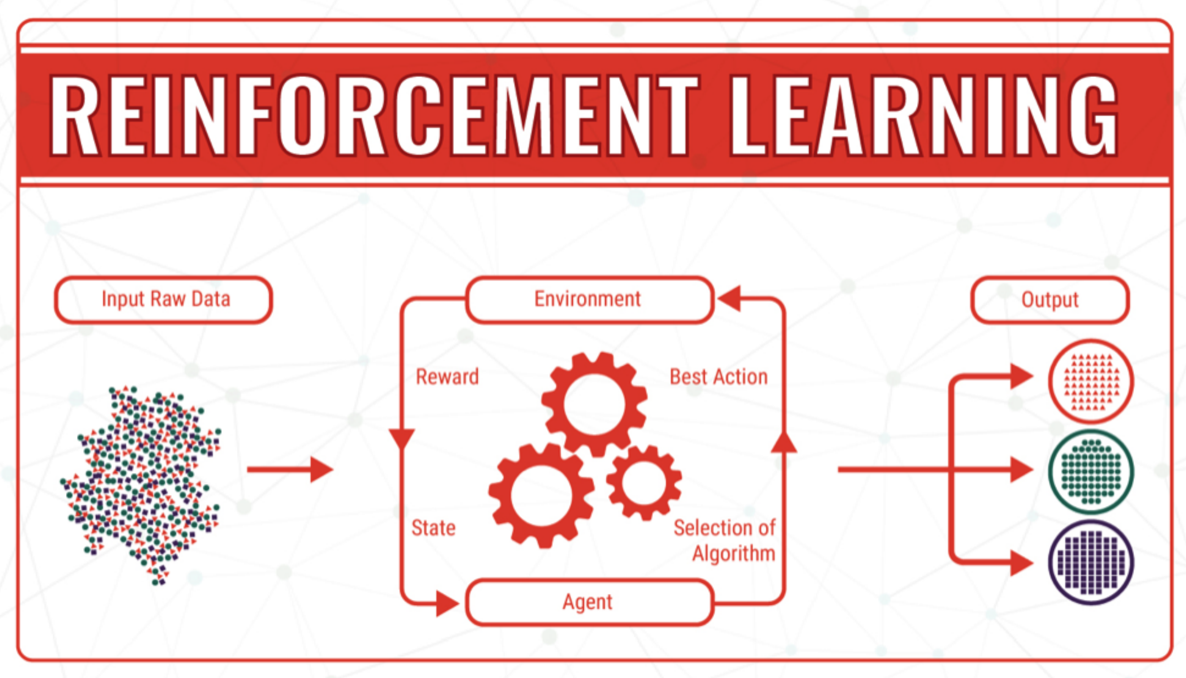
What makes unsupervised learning such an interesting area is that an overwhelming majority of data in this world is unlabeled. Having intelligent algorithms that can take our terabytes and terabytes of unlabeled data and make sense of it is a huge source of potential profit for many industries. That alone could help boost productivity in a number of fields.

Because unsupervised learning is based upon the data and its properties, we can say that unsupervised learning is data-driven. The outcomes from an unsupervised learning task are controlled by the data and the way it's formatted. Some areas you might see unsupervised learning crop up are:

- **Recommender Systems:** If you've ever used YouTube or Netflix, you've most likely encountered a video recommendation system. These systems are oftentimes placed in the unsupervised domain. We know things about videos, maybe their length, their genre, etc. We also know the watch history of many users. Taking into account users that have watched similar videos as you and then enjoyed other videos that you have yet to see, a recommender system can see this relationship in the data and prompt you with such a suggestion.
- **Buying Habits:** It is likely that your buying habits are contained in a database somewhere and that data is being bought and sold actively at this time. These buying habits can be used in unsupervised learning algorithms to group customers into similar purchasing segments. This helps companies market to these grouped segments and can even resemble recommender systems.
- **Grouping User Logs:** Less user facing, but still very relevant, we can use unsupervised learning to group user logs and issues. This can help companies identify central themes to

issues their customers face and rectify these issues, through improving a product or designing an FAQ to handle common issues. Either way, it is something that is actively done and if you've ever submitted an issue with a product or submitted a bug report, it is likely that it was fed to an unsupervised learning algorithm to cluster it with other similar issues.

Reinforcement Learning



Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning up is a bit murkier.

You can look at reinforcement learning as learning from mistakes. Place a reinforcement learning algorithm into any environment and it will make a lot of mistakes in the beginning. So long as we provide some sort of signal to the algorithm that associates good behaviors with a positive signal and bad behaviors with a negative one, we can reinforce our algorithm to prefer good behaviors over bad ones. Over time, our learning algorithm learns to make less mistakes than it used to.

Reinforcement learning is very behavior driven. It has influences from the fields of neuroscience and psychology. Let's look at teaching an agent to play the game Mario. For any reinforcement learning problem, we need an agent and an environment as well as a way to connect the two through a feedback loop. To connect the agent to the environment, we give it a set of actions that it can take that affect the environment. To connect the environment to the agent, we have it continually issue two signals to the agent: an updated state and a reward (our reinforcement signal for behavior). In the game of Mario, our agent is our learning algorithm and our environment is the game (most likely a specific level). Our agent has a set of actions. These will be our button states. Our updated state will be each game frame as time passes and our reward signal will be the change in score. So long as we connect all these components together, we will have set up a reinforcement learning scenario to play the game Mario. Some other real world examples of Reinforcement Learning are as follows:

- **Video Games:** One of the most common places to look at reinforcement learning is in learning to play games. Look at Google's reinforcement learning application, AlphaZero and AlphaGo which learned to play the game Go. Our Mario example is also a common example. Currently, I don't know any production-grade game that has a reinforcement learning agent deployed as its game AI, but I can imagine that this will soon be an interesting option for game devs to employ.
- **Industrial Simulation:** For many robotic applications (think assembly lines), it is useful to have our machines learn to complete their tasks without having to hardcode their processes. This can be a cheaper and safer option; it can even be less prone to failure. We can also incentivize our machines to use less electricity, so as to save us money. More than that, we can start this all within a simulation so as to not waste money if we potentially break our machine.
- **Resource Management:** Reinforcement learning is good for navigating complex environments. It can handle the need to balance certain requirements. Take, for example, Google's data centers. They used reinforcement learning to balance the need to satisfy our power requirements, but do it as efficiently as possible, cutting major costs.

Conclusion

Now that we've discussed the three different categories of machine learning, it's important to note that a lot of times the lines between these types of learning blur. More than that, there are a lot of tasks that can easily be phrased as one type of learning and then transformed into another paradigm.

For instance, take a recommender system. We discussed it as an unsupervised learning task. It can also easily be rephrased as a supervised task. Given a bunch of users' watch histories, predict whether a certain film should be recommended or not recommended. The reason for this is that in the end, all learning is learning. It's simply how we phrase the problem statement. Certain problems are more easily phrased one way or another.

That also highlights another interesting idea. We can blend these types of learning, designing components of systems that learn one way or another, but integrate together in one larger algorithm. An agent that plays Mario? Why not give it the supervised learning ability to recognize and label enemies? A system that classifies sentences? Why not give it the ability to capitalize on a representation of sentence meaning, learned through an unsupervised process? Want to group people in a social network into key segments and social groups? Why not add in a reinforcement process that refines the representation of a person so that we can more accurately cluster them?

Again, I think it is very important that we all understand a bit of machine learning, even if we will never create a machine learning system ourselves. Our world is drastically changing with machine learning becoming increasingly more prevalent in everything we use each day. Understanding even the fundamentals will help us to navigate this world, demystifying what can seem like a lofty concept and allowing us to better reason about the technology that we use.