Machine Learning

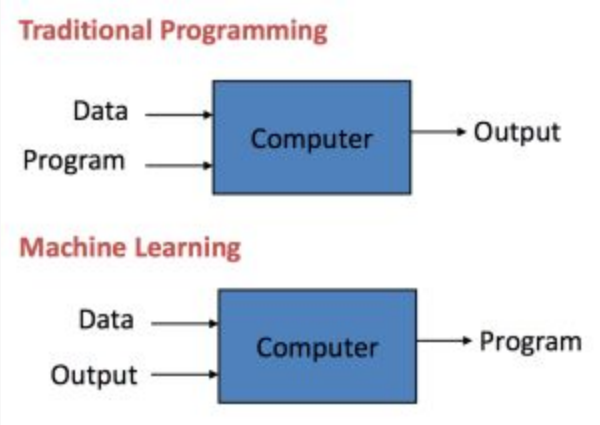
What is Machine Learning?

Machine Learning is getting computers to program themselves. If programming is automation, then machine learning is automating the process of automation.

Writing software is the bottleneck, we don't have enough good developers. Let the data do the work instead of people. Machine learning is the way to make programming scalable.

- **Traditional Programming**: Data and program is run on the computer to produce the output.
- **Machine Learning**: Data and output is run on the computer to create a program. This program can be used in traditional programming.

Machine learning is like farming or gardening. Seeds are algorithms, nutrients are data, the gardner is you and plants are the programs.



Traditional Programming vs Machine Learning

Applications of Machine Learning

Sample applications of machine learning:

- **Web search**: ranking page based on what you are most likely to click on.

- **Computational** biology: rational design drugs in the computer based on past experiments.
- **Finance**: decide who to send what credit card offers to. Evaluation of risk on credit offers. How to decide where to invest money.
- **E-commerce**: Predicting customer churn. Whether or not a transaction is fraudulent.
- **Space exploration**: space probes and radio astronomy.
- **Robotics**: how to handle uncertainty in new environments. Autonomous. Self-driving car.
- **Information extraction**: Ask questions over databases across the web.
- **Social networks**: Data on relationships and preferences. Machine learning to extract value from data.
- **Debugging**: Use in computer science problems like debugging. Labor intensive process. Could suggest where the bug could be.

What is your domain of interest and how could you use machine learning in that domain?

Key Elements of Machine Learning

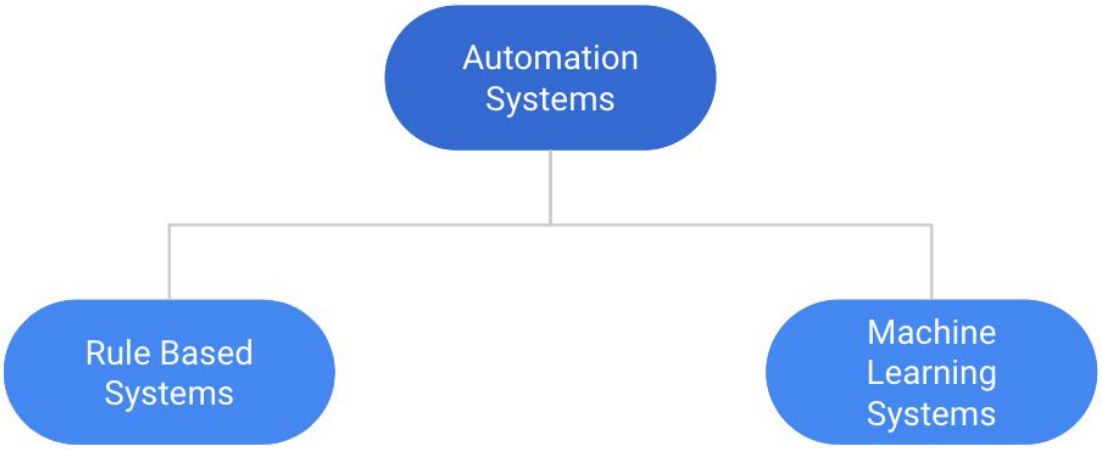
There are tens of thousands of machine learning algorithms and hundreds of new algorithms are developed every year.

Every machine learning algorithm has three components:

- **Representation**: how to represent knowledge. Examples include decision trees, sets of rules, instances, graphical models, neural networks, support vector machines, model ensembles and others.
- **Evaluation**: the way to evaluate candidate programs (hypotheses). Examples include accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.
- **Optimization**: the way candidate programs are generated known as the search process. For example combinatorial optimization, convex optimization, constrained optimization.

All machine learning algorithms are combinations of these three components. A framework for understanding all algorithms.

Introduction



Broadly speaking, automation is a technology which enables a process to be performed without human intervention. There are two types of automation systems

- Rule Based Systems
- Machine Learning Systems

A rule-based system represents knowledge in terms of a bunch of rules that tell you what you should do or what you could conclude in different situations. Again, like Data Science, the definition of Machine learning(ML) is up for debate. One of the widely accepted definitions has been provided by Arthur Samuels.

> Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. - Arthur Samuels (1959)

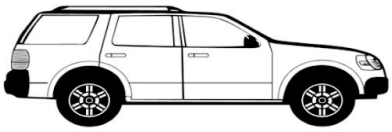
Before the advent of ML as a formal field of application and study, rule-based approaches were used to simulate intelligence. The `intelligence` of the computer system that played chess and defeated a grandmaster, Deep Blue, was essentially improved by a set of rules that were defined by other grandmasters of chess and improved search algorithms. So, essentially an expert defined a set of rules which helped automate some tasks and reduce human intervention. Thus, another name for these rule-based systems is `Expert Systems`.

Comparing Rule-Based and ML Systems

If we think a little deeper, traditional computer programs are rule-based systems at heart. So, let's design a rule-based system on how to recognize a car from a set of images. General rules for a car are:

- It has wheels
- It has headlight
- It has steering wheel

Then for a computer program, you can make the algorithm if it has wheels AND if it has headlights AND if it has a steering wheel. You will code it up and then the input arrives, as shown below, which breaks your existing system.



What if the car is at an angle in the picture? What if the image of the car is just a drawing? It would be really CUMBERSOME to list down all scenarios and all possible rules.

On the other hand, think of a scenario where you can just pass on the different images of the car to a program, along with some data to help identify the car and then allow the program to figure out the rest of the rules by itself. This is a typical ML program - which, when given enough data will generalize and figure out the rest of the rules by itself.

So, the difference between a traditional computer program and a Machine Learning program is:

A Traditional Algorithm is a clear set of instructions to which we give input and get a desired output. A rule-based approach.

An ML Algorithm - gives a set of data and let's the algorithm figure out patterns and rules to apply for future inputs.

Why Rule-Based Systems Fail

`Expert systems` worked well before because the data generated was very structured in nature and the scope was limited. Thus, it was possible to define rules by hand. As the size of data keeps increasing, the rules need to be modified constantly. With the advent of semi-structured and unstructured data, rule-based systems were no longer practical. It would be something like fixing bugs as the code keeps getting bigger. The advantage of ML is that it gets better as the size of data keeps increasing. In summary, the advantage of ML over expert systems is that it is scalable and generalizes well on data.

We have looked at the basic intuition of ML and also looked at it in contrast with expert systems. There are a lot of technical terms associated with Machine Learning, which we will introduce as and when the need arises.

Destroying the ML Misconceptions

Before we deep dive into the various aspects of Machine Learning, it is important to do a sanity check on the popular misconceptions surrounding it.

Myth 1: Autonomous Learning

For the machines to be successful and provide right insights on data, a lot of work needs to be done by humans on data to get it ready. True autonomous learning is not possible and humans cannot be completely removed from the equation. For example, look at the report in Financial Times on how self-driving cars are proving to be labor intensive.

Myth 2: Universal Application

> Surprisingly, despite AI's breadth of impact, the types of it being deployed are still extremely limited. Almost all of AI's recent progress is through one type, in which some input data (A) is used to quickly generate some simple response (B). - Andrew Ng 2016 (Source: HBR)

Nothing summarizes the response to the myth better than these words of Andrew Ng, one of the pioneers in ML. Read more on his thoughts on this subject here. Also, ML cannot be applied effectively when collection of data is very difficult like in health domains.

On the other side of the coin, a lot of problems in the industry can be solved through simple data summarization and analysis and don't need AI. Why use complicated statistical models to solve problems that could be solved by simple rules?

Myth 3: Transformational Insights

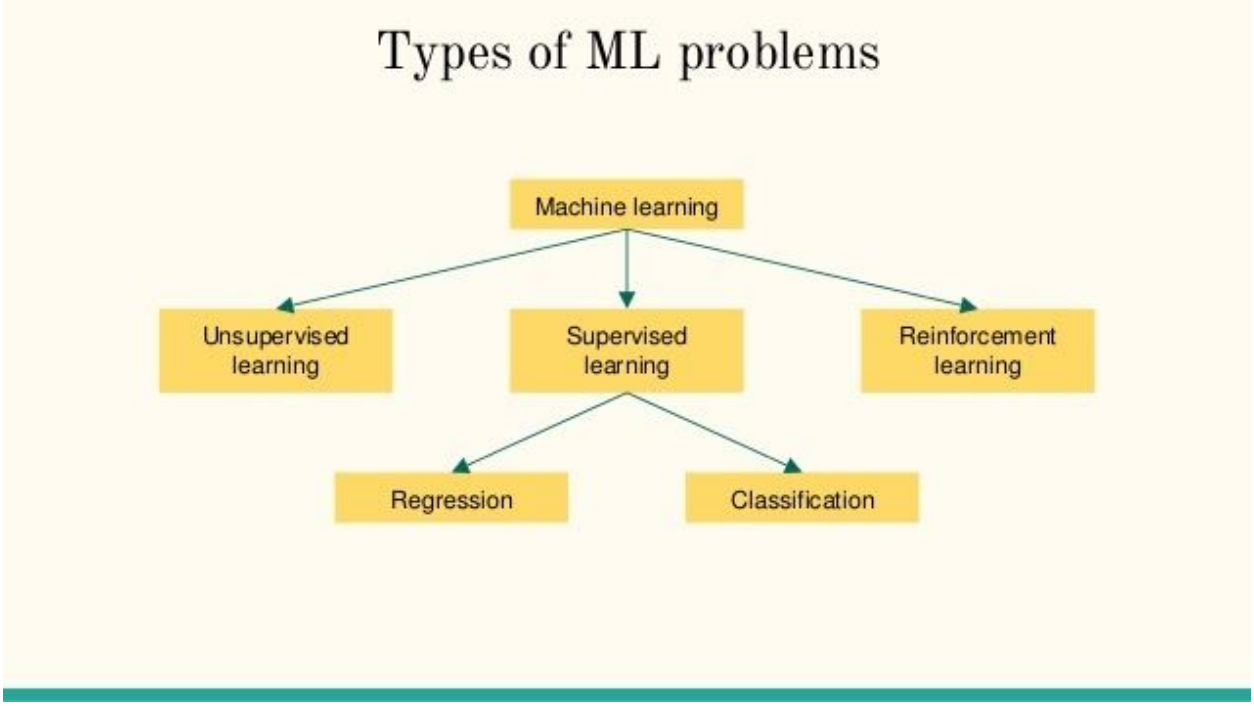
A popular myth is that once you feed the algorithm data, actionable insights will emerge like the pot of gold at the edge of the rainbow. It is exactly that - a mirage. For every data generated, defining the problem to solve is very hard and it is a continuous iterative process of refining data and rebuilding models.

Myth 4: Superhuman Intelligence

Machines are as intelligent as the humans who programmed it. Despite all the advancement, machines don't have common sense and cannot be taught the same. Don't worry - your smartphone won't take over the world and enslave you. EVER!

Now that you have overcome the fear of machines taking over the world, let's look formally at the different types of ML.

Broad Hierarchy of ML Algorithms



While there are different classes of ML algorithms, they differ in the way the input is presented to the algorithm and the output that is expected. For supervised ML, input is tagged with expected outputs and we try to predict the outputs for future inputs too. Some examples of supervised learning are as follows:

INPUT A	RESPONSE B	APPLICATION
Picture	Are there human faces? (0 or 1)	Photo tagging
Loan application	Will they repay the loan? (0 or 1)	Loan approvals
Ad plus user information	Will user click on ad? (0 or 1)	Targeted online ads
Audio clip	Transcript of audio clip	Speech recognition
English sentence	French sentence	Language translation
Sensors from hard disk, plane engine, etc.	Is it about to fail?	Preventive maintenance
Car camera and other sensors	Position of other cars	Self-driving cars

Source: Andrew Ng, HBR

- Formally, the main task of the ML algorithm is to learn a mapping function from input to output variable/variables.
- $\mathbf{Y = F(x)}$

$$\mathbf{Y=F(x)}$$

Train/Test Data

Let's take a simple scenario of teaching a child multiplication tables. You have taught them everything from 1x1 and 9x9. Now, how would you test whether the child has truly understood multiplication or simply memorized the tables? A good way to do that would be to give unseen data, like 11x12. Here we can say that 1x1 to 9x9 is the training data and 11x12 is the test data.

Similarly, in ML too, we would like to see if the ML algorithm has truly learned the relationship between input and output. To do that, we divide the data into two parts - training data and test data. We use training data to learn the given model. And then, we use the test data to evaluate the model. What `learning a model` actually means is that we will study it in other concepts down the line. Broadly, there are two types of supervised learning tasks -- Classification and Regression.

The main difference between classification and regression is the form of the output variable. If the output variable is discrete, it is classification, and if it is continuous, it is regression. You can think of discrete data as something that is counted, and continuous data as something that is measured. Discrete data refers to things like movie genres, categories of mail (spam/not spam) and so on. Continuous data refers to data like the height of a person, stock price, price of a house, and so on.

Classification

- It involves finding a set of categories a new data point it belongs to. The output value is a class or a category.
- Examples:
    - Deciding if the image - a cat or dog
    - Deciding if an email is spam or not
    - Predicting whether a patient has cancer or not
    - Predicting whether the value of a stock will rise or drop

Important

- Qualitative Output
- Predefined Classes

Regression

- It involves predicting a continuous value, given a set of inputs. The output value is continuous.
- Examples:
    - Predicting the stock price given different factors
    - Predicting the sales of a company for a given year
    - Predicting runs/goals scored by a sports team
    - Assigning credit rating

Important

- Quantitative Output
- Previous input-output observations