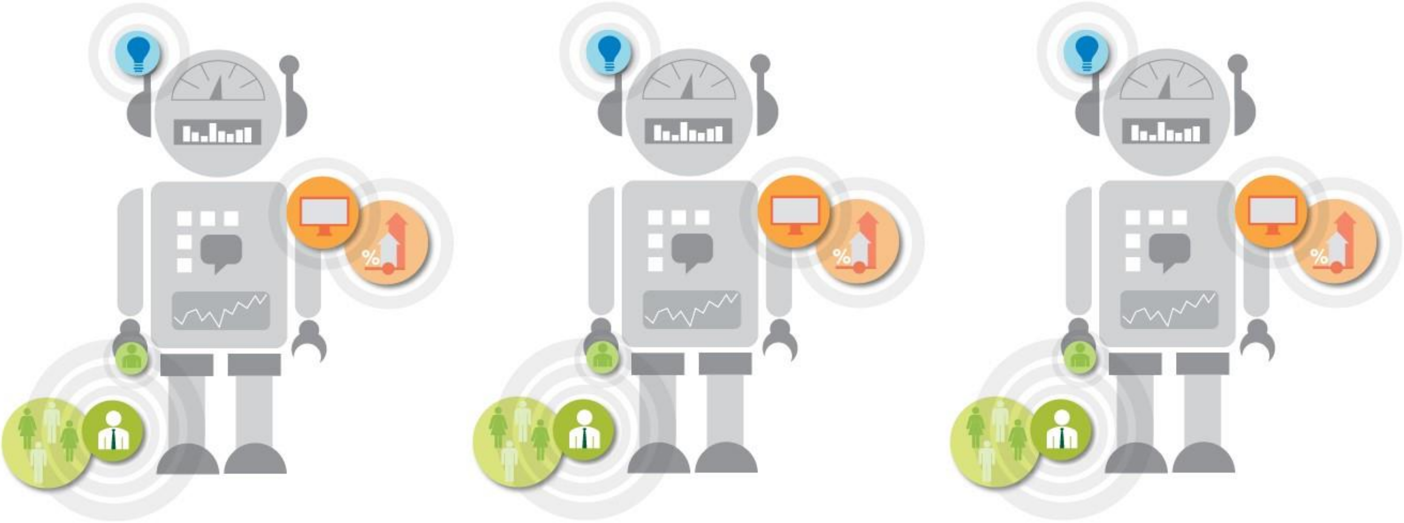
Getting Started with Machine Learning (ML)

Rukshan Manorathna
University of Colombo
14 PUBLICATIONS **7** CITATIONS

Getting Started with Machine Learning (ML)

A disruptive technology which sweeps away traditional programming in certain cases…

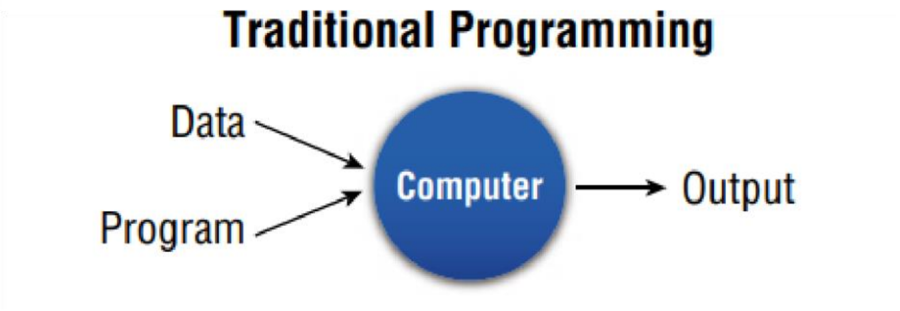
Rukshan Pramoditha
Dec 25, 2019 · 9 min read



After a very long time since the first article **Getting Started with Data Science** was published in **Data Science 365**, today, I meet you with a new topic in the field of Data Science. If you haven’t read my first article yet, I recommend you to read it before reading this one. Without further delay, welcome to *Getting Started with Machine Learning (ML)*!

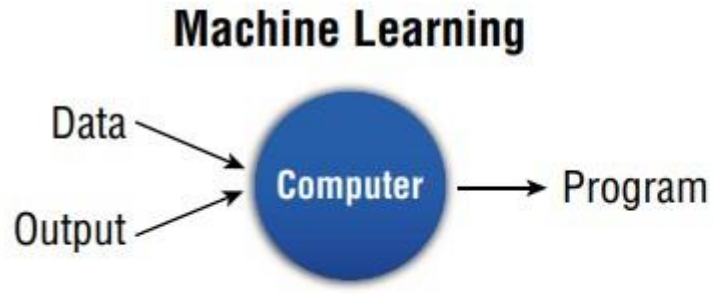
Introduction

Machine Learning (ML) is a collection of algorithms & techniques used to build systems that learn from data. These systems are then able to perform predictions by finding patterns in data. Machine learning is a disruptive technology which sweeps away traditional programming in certain cases. ***What is the difference between machine learning & traditional programming?***



In traditional programming, the data & the program produce the output. For example, when performing some accounting tasks, the program takes in the data (sales records, inventory lists, etc.) & calculates your profits or losses. It will also give some nice & fanciful charts showing your sales performance.

Traditional programming is a manual process — meaning a person (programmer) creates the program by manually formulating or coding the rules.



In machine learning, the data & the output produce the program. You take both data & output & use them to derive a set of rules to make predictions. It means that you may use this model to predict the most popular items that will sell next year.

In machine learning, the algorithm automatically formulates the rules from the data.

ML consists of the following disciplines:

- Scientific computing (e.g. Python or R programming)
- Mathematics
- Statistics

When using ML, a data scientist needs to know about the followings:

- Which method of machine learning will best help in completing the given task
- How to apply that method

The knowledge that how that method works is optional.

Why machine learning?

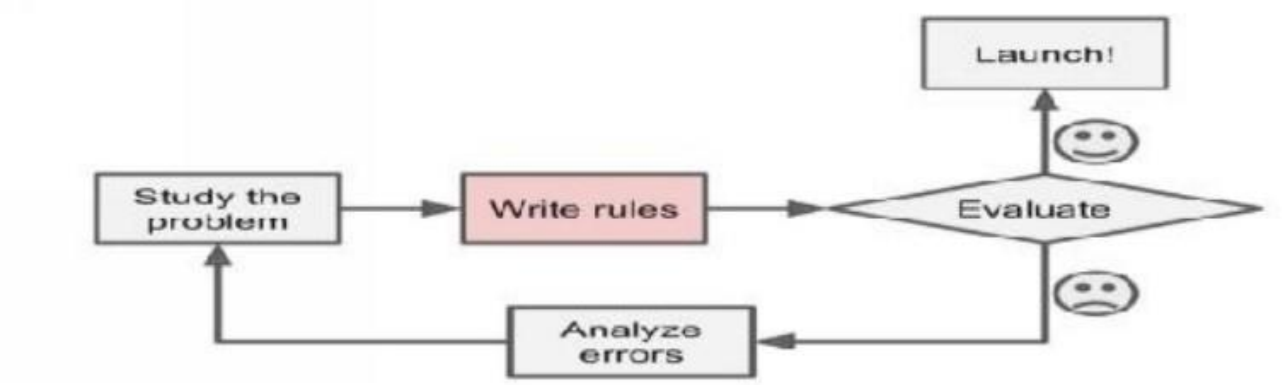
Consider an event of creating a spam e-mail filter program ***without using ML***. The programmer writes that program by following the steps in a certain order.

Step 1: Identify how spam e-mails look like. (Words: “debit card”, “free”, “for you”, etc.)

Step 2: Write an algorithm to detect the patterns that you’ve seen.

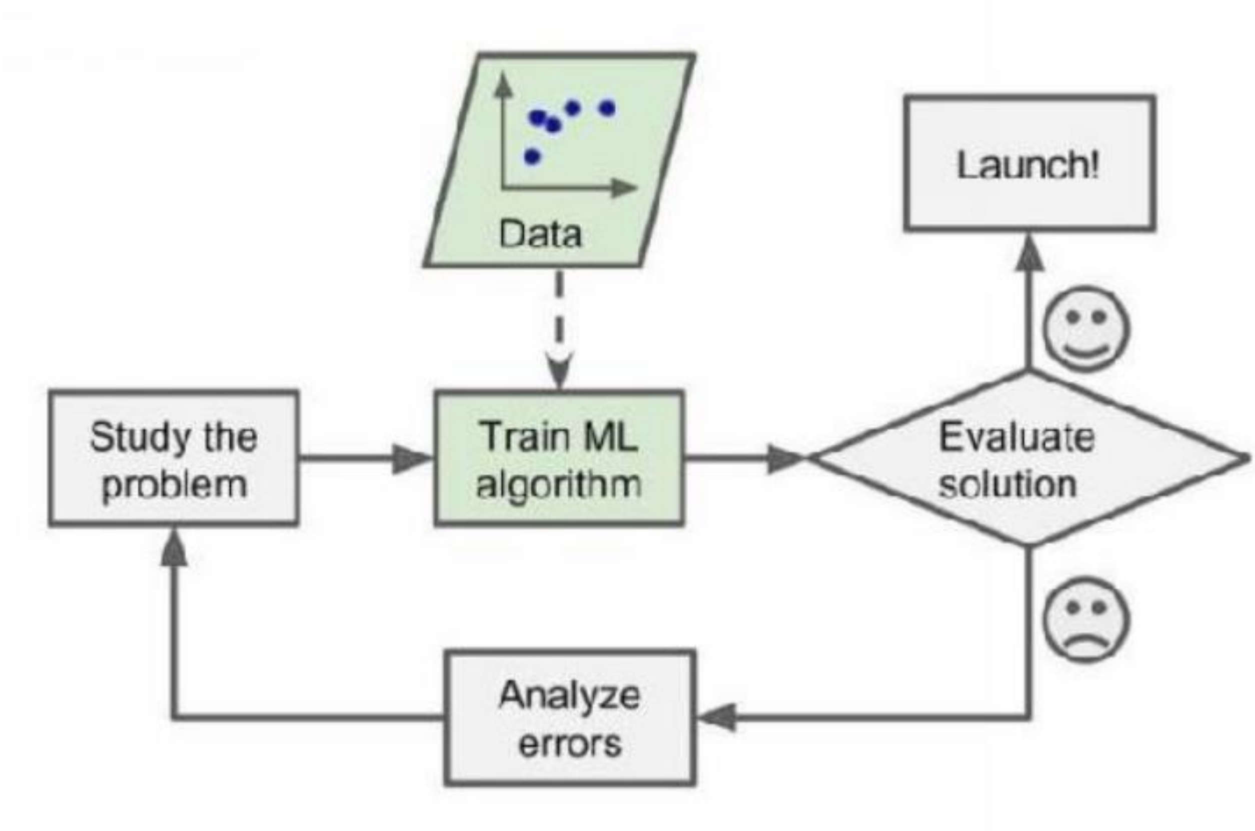
Step 3: Finally, you’d test the program & then make changes to the program until the results are good enough.

After that, the software would flag emails as spam if a certain number of those patterns are detected.



In your algorithm, you **write rules** in your program to detect the patterns. If this is a case that requires a long list of rules to find the solution, it soon becomes difficult for a human to accurately code the rules. You can use ML to effectively solve this problem.

What will happen if the e-mail senders change their e-mail templates so that a word like “4U” now instead of “for you”? The program using traditional techniques would need to be updated manually. On the other hand, a program using ML techniques will automatically detect this change & will adapt to new data.



When should you use machine learning?

ML is not a solution for every type of problem. There are certain cases where solutions can be developed without using ML techniques. For example, you don't need ML if you can determine a target value by using simple rules & computations. If you implement ML techniques for such problems, they will be getting more complex & you will need more computer power to get the solutions.

However, there are some situations where you need to use ML.

- **Very complex problems for which there is no solution with a traditional approach:** You need to use a data-driven approach to get the solution. For example, speech recognition: When you say "one" or "two", the program should distinguish the difference. You will need to develop an algorithm that measures sound.

- **Non-stable environments:** Machine learning software can adapt to new data.

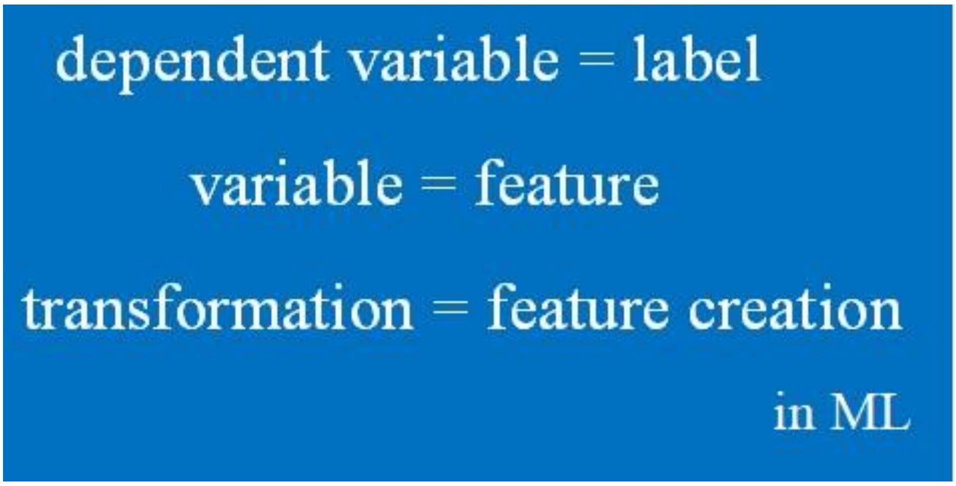
- **When you have a problem that requires many long lists of rules to find the solution:** When rules depend on too many factors, it soon becomes difficult for a human to accurately code the rules. You can use ML to effectively solve this problem.

- **You cannot scale:** ML solutions are effective at handling large-scale problems. For example, a spam e-mail filter program: There are millions of emails which are needed to check in a single day! It cannot be done manually.

A few of ML terminology

There is a long list of ML terminology. Here I list some terms which are mostly used. Others will be given in relevant parts as we progress.

- In ML, a target is called a **label**. In statistics, a target is called a **dependent variable**.
- A variable in statistics is called a **feature** in ML.
- A transformation in statistics is called a **feature creation** in ML.



dependent variable = label

variable = feature

transformation = feature creation

in ML

Types of machine learning

There are 2 main types of Machine Learning which are based on their learning styles. They are:

- **Supervised Learning**
- **Unsupervised Learning**

In addition to these types, there are other types as well:

- Semi-supervised Learning
- Reinforcement Learning
- Batch Learning
- Online Learning
- Instance-based Learning
- Model-based Learning

The criteria for the above classification are:

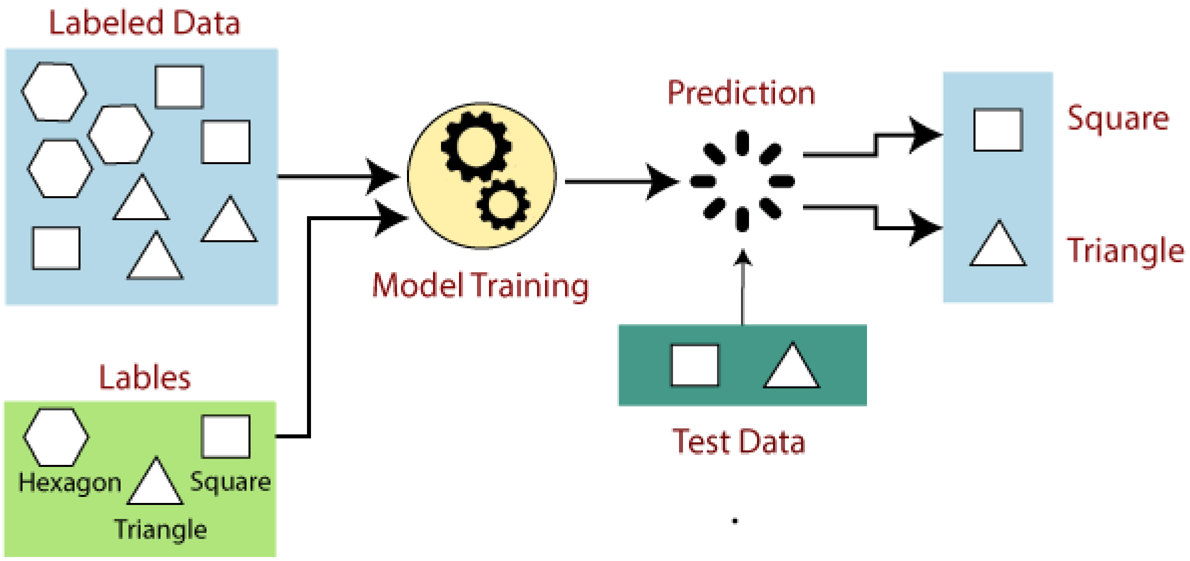
- **According to the type & amount of human supervision needed during the training:** Supervised Learning, Unsupervised Learning, Semi-supervised Learning & Reinforcement Learning are classified based on this criterion.
- **If they can learn incrementally**
- **If they work simply by comparing new data points to find data points or can detect new patterns in the data & then will build a model**

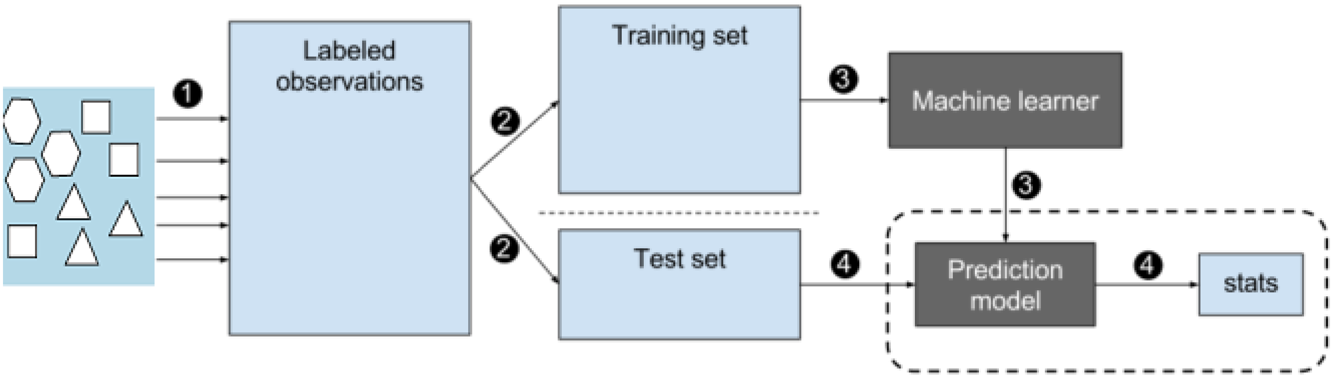
Supervised vs unsupervised learning

Supervised Learning

In supervised learning, you supervise the learning process. You train the algorithm using **labelled** data. You can correct your algorithm if it makes a mistake in giving you the answer in the learning process. This can be compared to learning which takes place in the presence of a supervisor or a teacher.

In supervised learning, both input & output variables are given. This can be explained mathematically. You have input variables (called **x**) & an output variable (called **y**) & you use an algorithm to learn the mapping function of from the input to the output, y=f(x). **In supervised learning, the goal is to determine the mapping function so well that when you have new input data (x), you can predict the output for that data**. If the mapping is correct, the algorithm has successfully learned. Else, you make the necessary changes to the algorithm so that it can learn correctly. In simple words, what we do in supervise learning is that we use a labelled dataset to obtain a new label for unlabelled data.



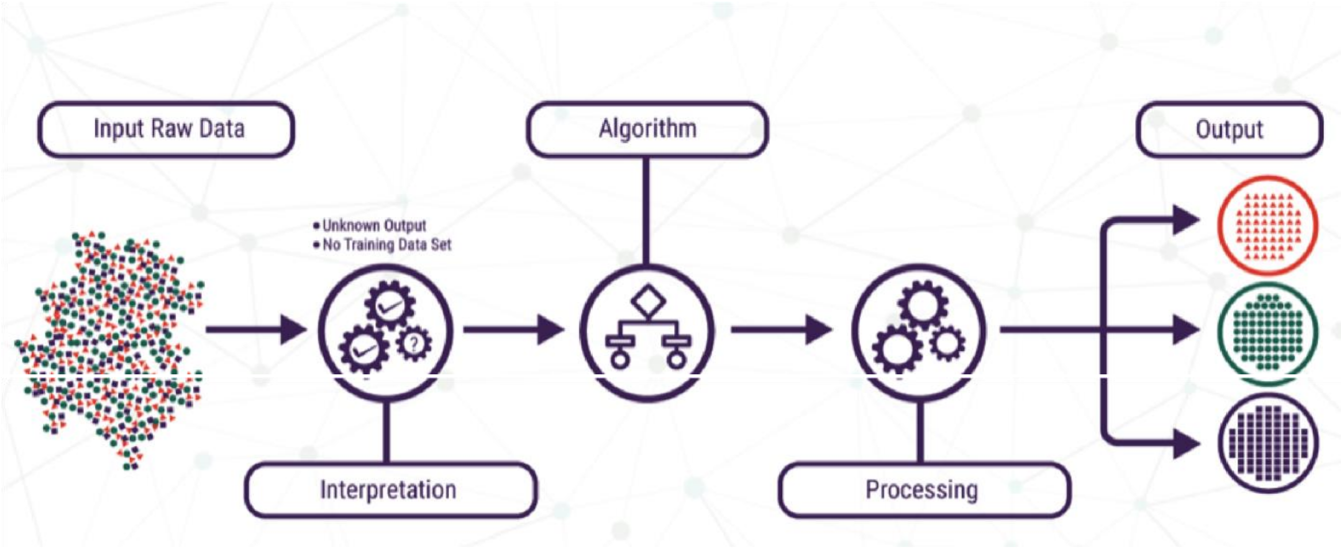


There are two phases: **Training Phase** & **Testing Phase**. In the training phase, you take a randomly selected specimen of geometric shapes (**training data**) & label them accurately. Then you make a table of all the characteristics (**features**) of each shape. You feed this data to the machine learning algorithm & it learns a model (called ***prediction model***). In the testing phase, you input a shape (**test data**) which was also labelled. The prediction model which was created earlier will give the correct label for that shape. If the output is correct, the algorithm has successfully learned. Else, you make the necessary changes to the algorithm so that it can learn correctly.

Classification and **Regression** are two types of supervised learning. Supervised learning is a highly accurate method. The main drawback is that classifying big data can be a real challenge.

Unsupervised Learning

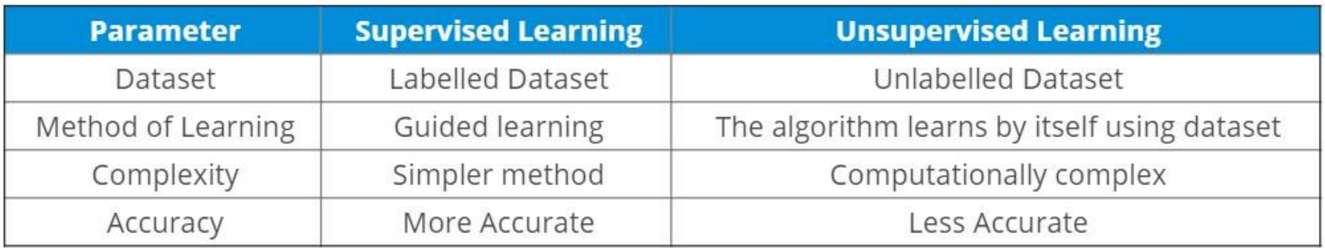
In unsupervised learning, you do not need to supervise the learning process. Instead, you need to allow the model to work on its own to discover hidden patterns in data. This can be compared to the learning process of a student who has textbooks and all the required material to study but has no teacher to guide so that he will have to learn by himself.



In unsupervised learning, only input data will be given. That data does not have any sort of labels. **The goal is to find the hidden patterns or the underlying structure in the given input data in order to learn about the data**. Unsupervised learning helps you to find features which can be useful for categorization. It can also help to detect anomalies and defects in the data which can be taken care of by us.

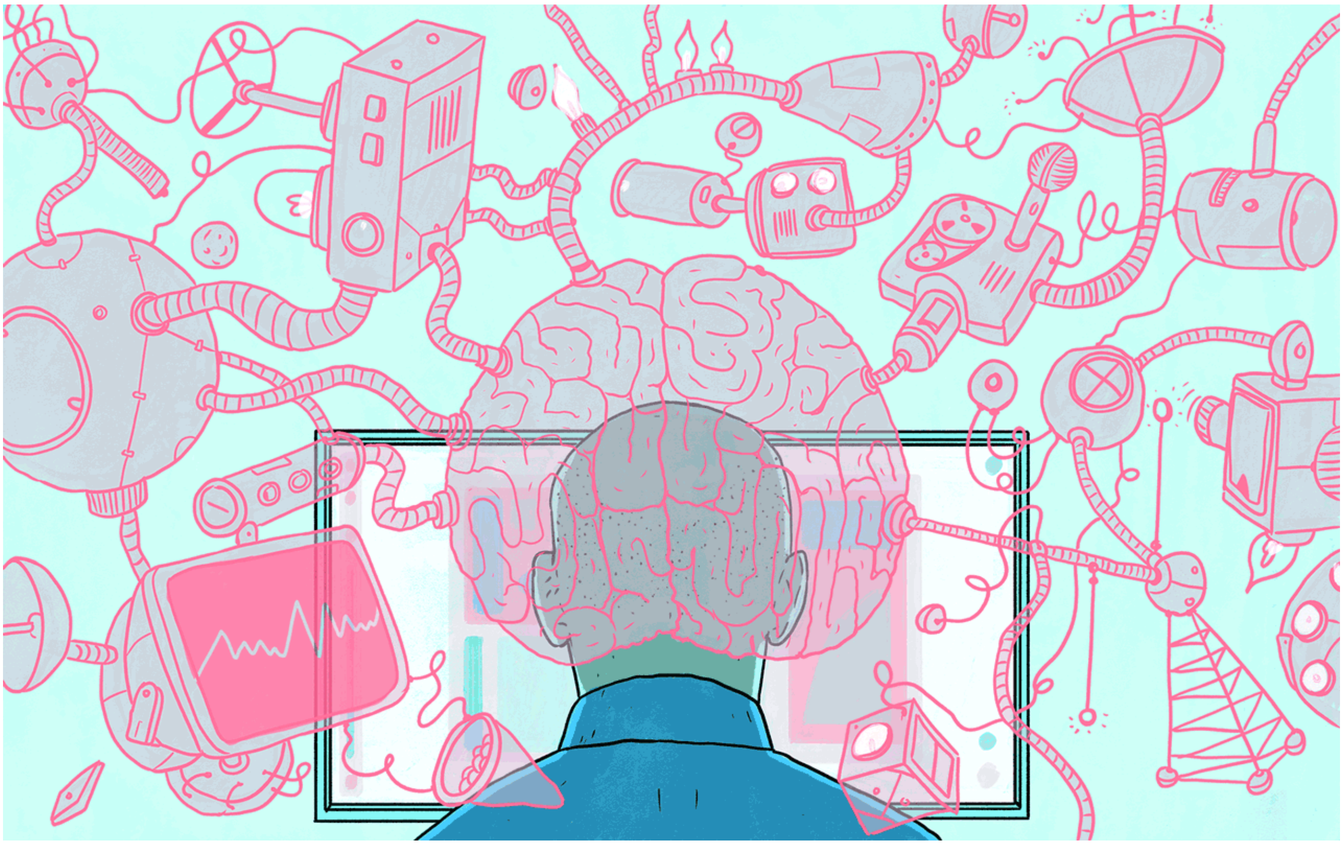
Clustering and **Association** are two types of unsupervised learning. Unsupervised learning is less accurate & computationally complex when compared to supervised learning.

As a summary, we can compare supervised & unsupervised learning methods as follows.

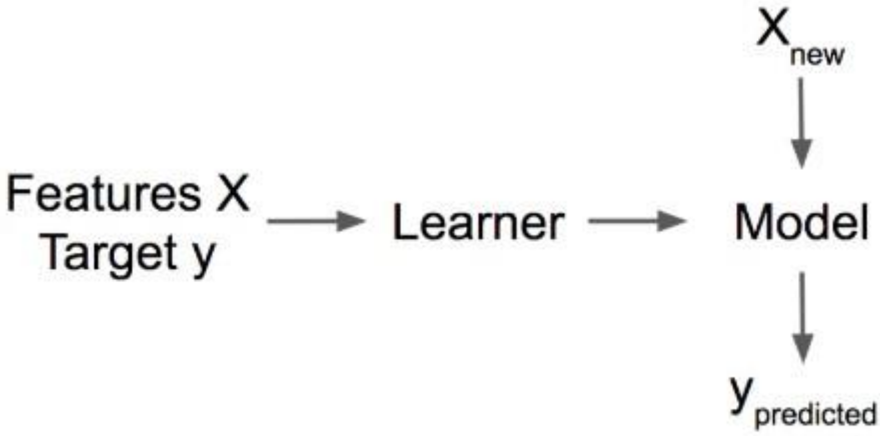


Parameter	Supervised Learning	Unsupervised Learning
Dataset	Labelled Dataset	Unlabelled Dataset
Method of Learning	Guided learning	The algorithm learns by itself using dataset
Complexity	Simpler method	Computationally complex
Accuracy	More Accurate	Less Accurate

Machine learning algorithms



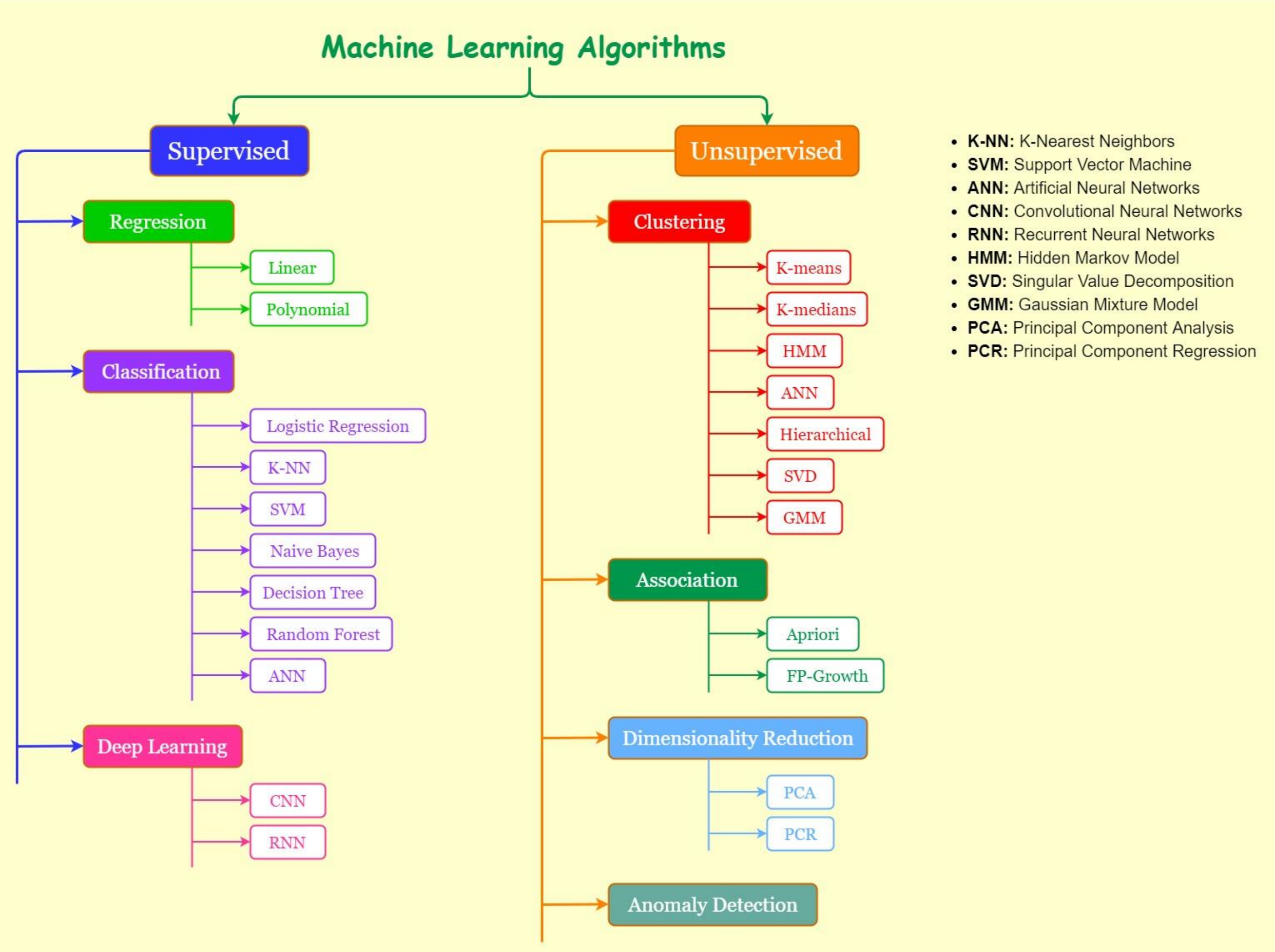
An **algorithm** is a set of rules that a machine follows to achieve a particular goal. These rules are in a certain order. A **learner** or **machine learning algorithm** is a set of rules used to learn a machine learning model from data. ML algorithms are given general guidelines that define the model, along with data while classical algorithms are given exact & complete rules to finish a task. An ML algorithm can accomplish its task when the model has been adjusted with respect to the data. We have to **fit the model on the data** or **the model has to be trained on the data**. ML algorithms are different from classical algorithms in a way that they automatically learn from the data you provide.



Some useful definitions of commonly used terms related to machine learning algorithms are:

- **Dataset:** A table with the data from which the machine learns. The dataset contains the features and the target to predict.
- **Instance:** A row in the dataset. Other names for "instance" are data point, observation. An instance consists of the feature values x(i) & if known, the target outcome y(i).
- **Feature:** An input used for the machine learning algorithm. A feature is a column in the dataset. The matrix with all the features is called X for a single instance. The vector of a single feature for all instances is x(j).
- **Target:** The information that the machine learns to predict. In mathematical formulas, the target is usually called y. In statistics, it is called a dependent variable.
- **Prediction:** The target value that the machine learning model "guesses" based on the given features.

Machine learning algorithms fall into two broad categories which are **supervised learning algorithms** & **unsupervised learning algorithms**, although there are other categories such as semi-supervised learning algorithms, reinforcement learning algorithms, etc. The following chart shows some of the commonly used algorithms which are classified under ***supervised*** & ***unsupervised***. Note that some of them can be in both supervised & unsupervised categories, although they are listed under one category.


Machine Learning Algorithms
Supervised
Regression
Linear
Polynomial
Classification
Logistic Regression
K-NN
SVM
Naive Bayes
Decision Tree
Random Forest
ANN
Deep Learning
CNN
RNN
Unsupervised
Clustering
K-means
K-medians
HMM
ANN
Hierarchical
SVD
GMM
Association
Apriori
FP-Growth
Dimensionality Reduction
PCA
PCR
Anomaly Detection
K-NN: K-Nearest Neighbors
SVM: Support Vector Machine
ANN: Artificial Neural Networks
CNN: Convolutional Neural Networks
RNN: Recurrent Neural Networks
HMM: Hidden Markov Model
SVD: Singular Value Decomposition
GMM: Gaussian Mixture Model
PCA: Principal Component Analysis
PCR: Principal Component Regression

Copyright: *Rukshan Pramoditha*

What is next?

This is just an introductory article for ML. This article lays a good foundation for ML & motivates you to learn more about ML. The next big part is to learn machine learning algorithms. Learning theoretical parts of these algorithms is not just enough. We also want to learn how to implement these algorithms using Python or R programming. Here I use Python. (*Selecting Python or R for doing Data Science & ML should be done by conducting thorough research. I personally selected Python by comparing many factors between the two languages*.)

The next article will be *logistic regression under classification algorithms in ML*. An article series about Python programming should also be written parallelly with the ML article series.

One last key point

My success will not be possible without your feedback. So please don’t hesitate to give me feedback. Write them in the comment section of this article or just drop a message at *rpromoditha@gmail.com*.

Thank you for reading! Next time, I will meet you with another ML article. Goodbye for now!

Written by: Rukshan Pramoditha

Data Science 365,
Bring data into actionable insights.