# INTRODUCTION TO MACHINE LEARNING

**By**

Md. Zubair
Lecturer Department of CSE, Uttara University

Email: md.zubair@uttarauniversity.edu.bd
Blog: https://medium.com/@mzh706
Quora Space: https://abcofdatascienceandml.quora.com/

# Machine Learning Introduction

When most people hear “Machine Learning,” they picture a robot: a dependable butler or a deadly Terminator, depending on who you ask. But Machine Learning is not just a futuristic fantasy; it’s already here. In fact, it has been around for decades in some specialized applications, such as Optical Character Recognition (OCR). But the first ML application that really became mainstream, improving the lives of hundreds of millions of people, took over the world back in the 1990s.

This class introduces a lot of fundamental concepts (and jargon) that every data scientist should know by heart. It will be a high-level overview all rather simple, but you should make sure everything is crystal clear to you before continuing on to the rest of the concepts. So grab a coffee and let’s get started!

# What Is Machine Learning?

Machine Learning is the science (and art) of programming computers so they can *learn from data*.

Here is a slightly more general definition:

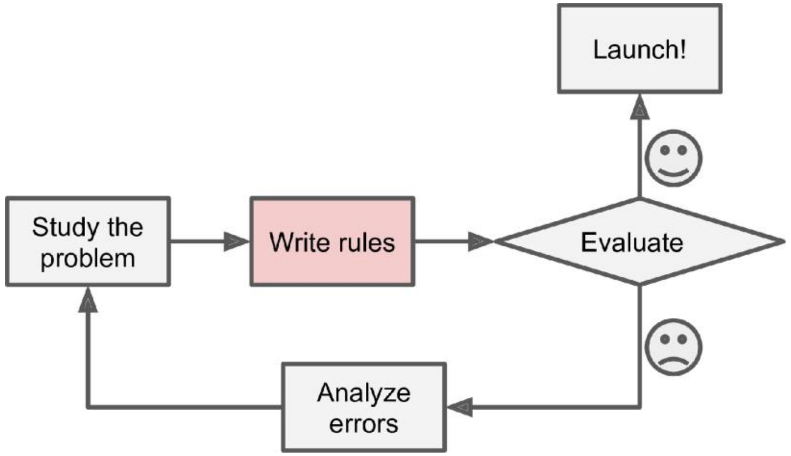
> *[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.*
>
> —Arthur Samuel, 1959

And a more engineering-oriented one:

> *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*
>
> —Tom Mitchell, 1997

# Why Use Machine Learning?

Consider how you would write a spam filter using traditional programming techniques



1. First you would consider what spam typically looks like. You might notice that some words or phrases (such as "4U," "credit card," "free," and "amazing") tend to come up a lot in the subject line. Perhaps you would also notice a few other patterns in the sender's name, the email's body, and other parts of the email.
2. You would write a detection algorithm for each of the patterns that you noticed, and your program would flag emails as spam if a number of these patterns were detected.
3. You would test your program and repeat steps 1 and 2 until it was good enough to launch.

**Since the problem is difficult, your program will likely become a long list of complex rules—pretty hard to maintain.**

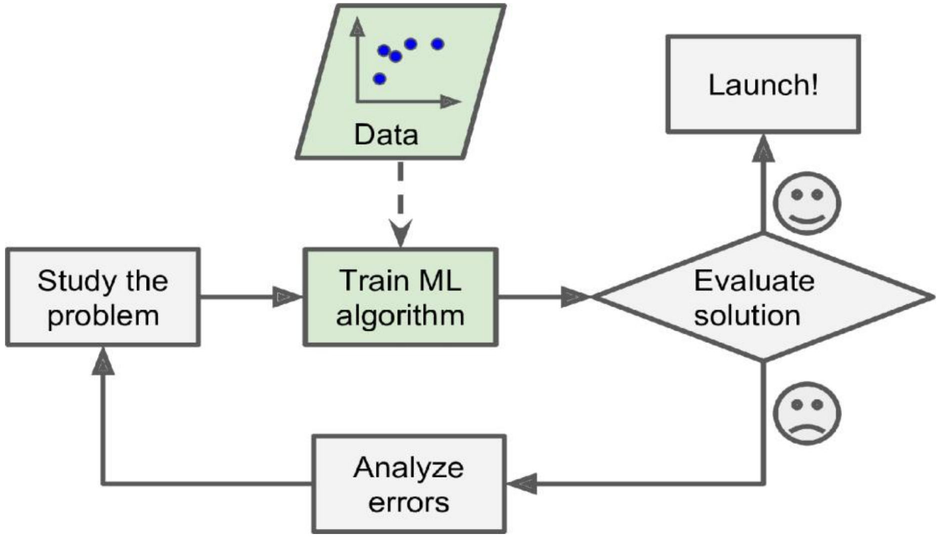


*Figure- The Machine Learning approach*

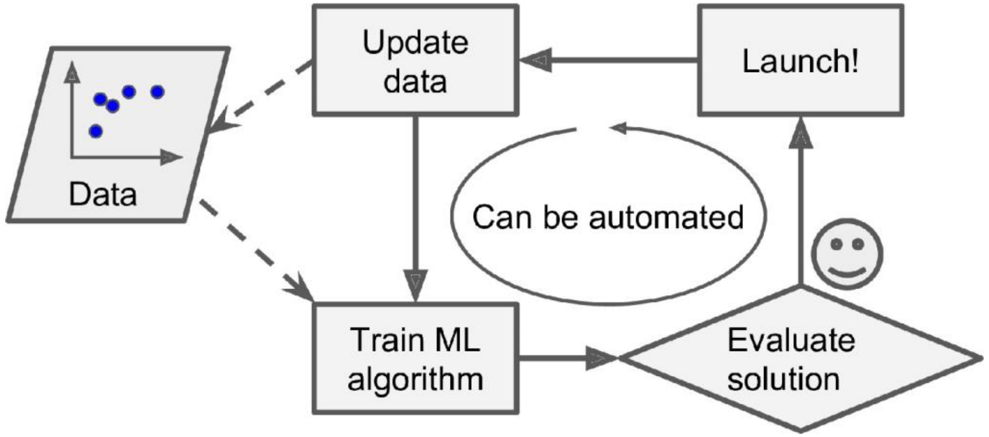


*Figure- Automatically adapting to change*

Suppose, in case of spam filtering technique, the spammer notice that the email containing “4U” is spam. They might start writing “For U” instead. A spam filter using traditional programming techniques would need to be updated to flag “For U” emails. If spammers keep working around your spam filter, you will need to keep writing new rules forever.

In contrast, a spam filter based on Machine Learning techniques automatically notices that “For U” has become unusually frequent in spam flagged by users, and it starts flagging them without your intervention

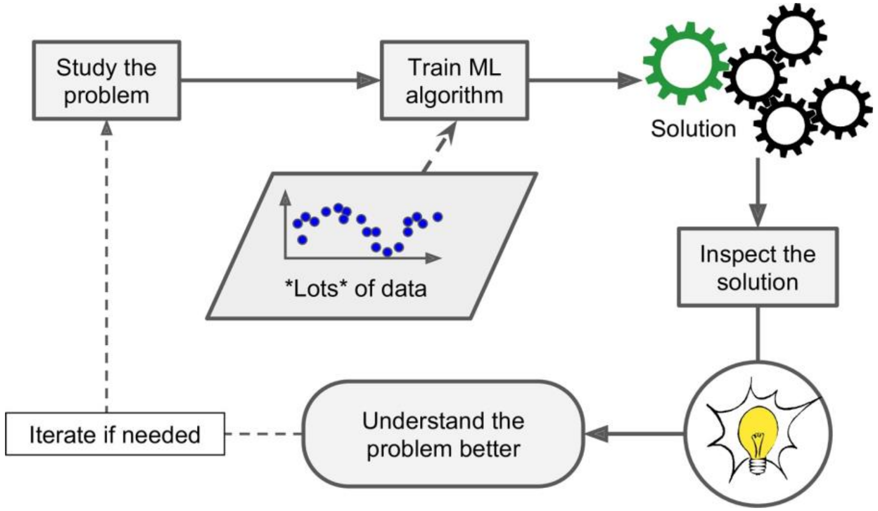


*Figure - Machine Learning can help humans learn*

To Summarize, Machine Learning is great for:

- Problems for which existing solutions require a lot of fine tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better tan the traditional approach.

- Complex problems for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution.

- Fluctuating environments: a machine Learning system can adapt to new data.

- Getting insights about complex problems and large amounts of data.

## Examples of Applications

- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans
- Automatically classifying news articles
- Automatically flagging offensive comments on discussion forums
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Forecasting your company's revenue next year, based on many performance metrics
- Making your app react to voice commands
- Detecting credit card fraud
- Segmenting clients based on their purchases so that you can design a different marketing strategy for each segment
- Representing a complex, high-dimensional dataset in a clear and insightful diagram
- Recommending a product that a client may be interested in, based on past purchases
- Building an intelligent bot for a game

**This list could go on and on, but hopefully it gives you a sense of the incredible breadth and complexity of the tasks that Machine Learning can tackle, and the types of techniques that you would use for each task.**

## Types of Machine Learning Systems

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories, based on the following criteria

- Whether or not they are trained with human supervision (**supervised, unsupervised, semisupervised, and Reinforcement Learning)**
- Whether or not they can learn incrementally on the fly (**online versus batch learning**)
- Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (**instance-based versus model-based learning**)

# Supervised Learning

In *supervised learning*, the training set you feed to the algorithm includes the desired solutions, called ***labels/target value.***

***There are two types of supervised learning algorithms.***

1. ***Classification***
2. ***2. Regression***

## Classification

If the labelled/target value is nominal then it is a classification problem.

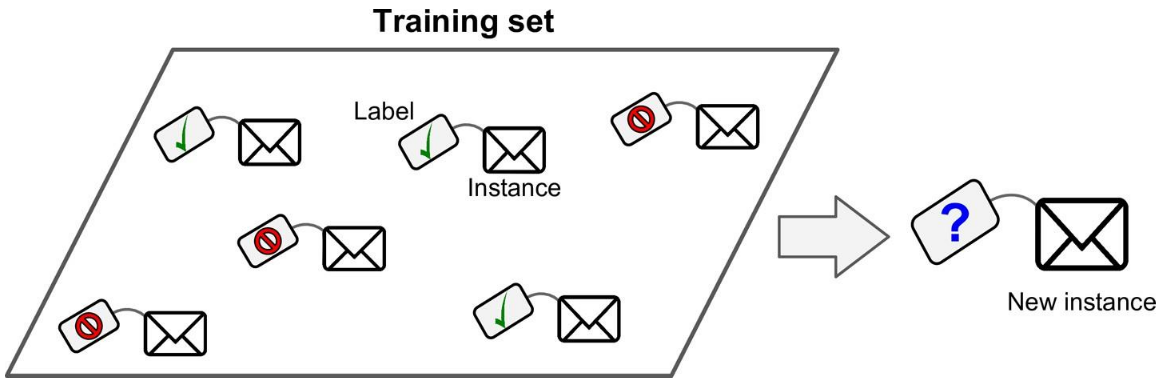


*Figure: A labeled training set for spam classification (an example of supervised learning)*

**Regression**

If the target value for the problem is continuous  value then the problem is regression.

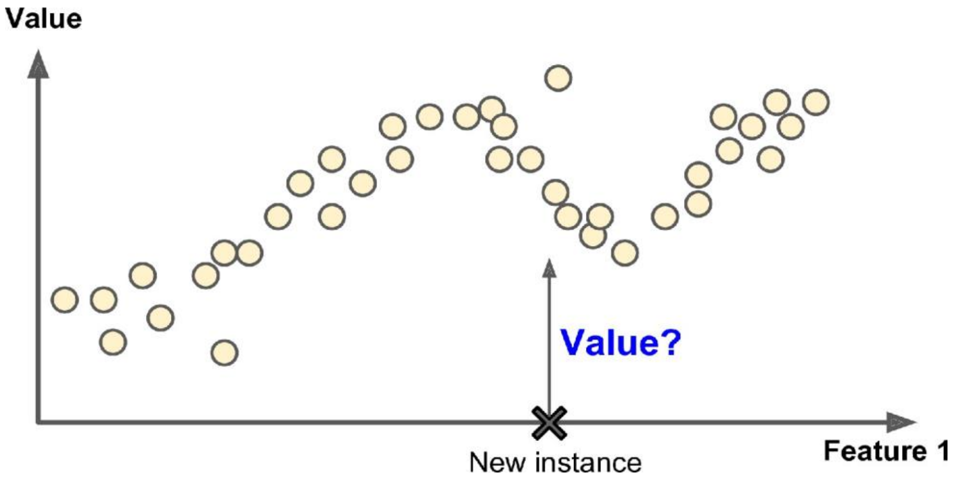


*Figure. A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)*

Here are some of the most important supervised learning algorithms

- K-Nearest Neighbour (Classification and regression)
- Linear Regression (regression)
- Logistic Regression (classification)
- Support Vector Machine (SVM) (Classification and regression)
- Decision Tree and Random Forests (classification and regression)
- Neural Networks (classification and regression)

## Unsupervised learning

In *unsupervised learning*, as you might guess, the training data is unlabeled. The system tries to learn without a teacher.

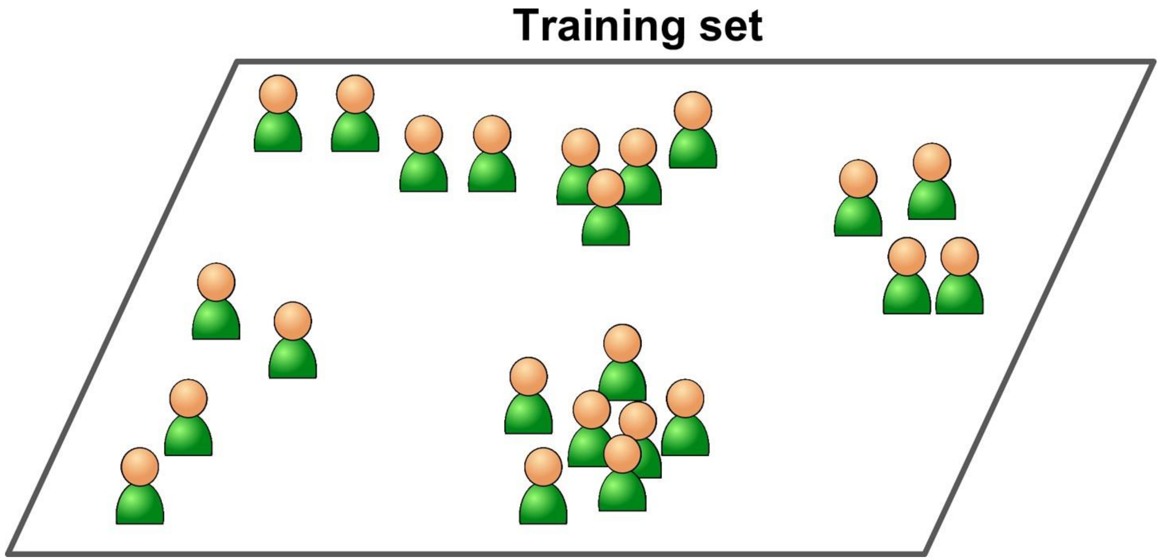


Figure- Unlabelled training set

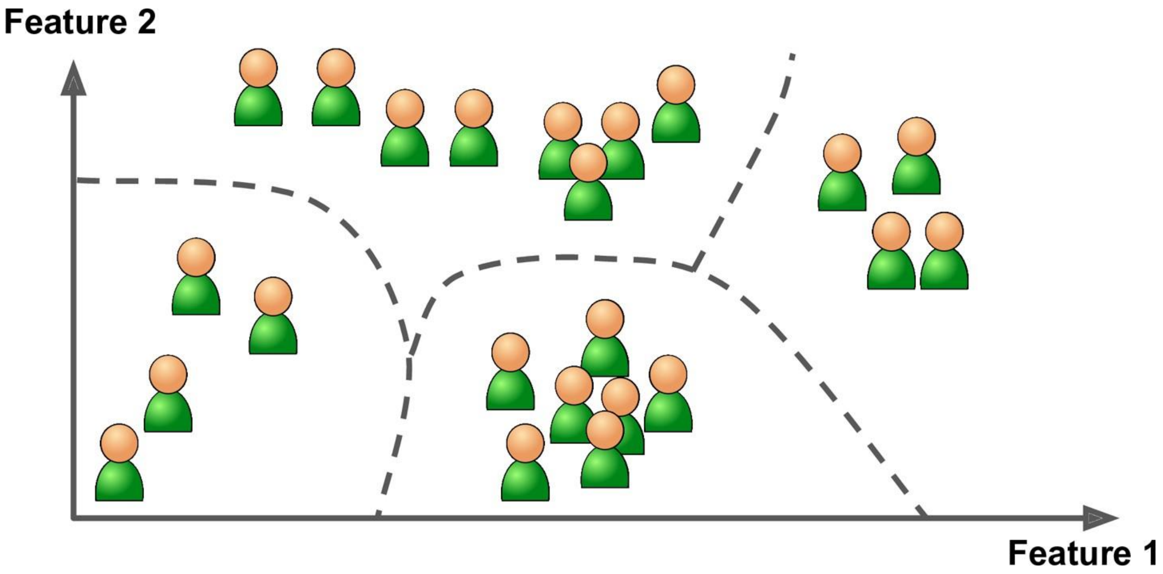


Figure- Clustering

## Here are some of the most important unsupervised learning algorithms

Clustering

- K-Means
- DBSCAN
- Hierarchical Cluster Analysis (HCA)

Anomaly detection and novelty detection

- One –class SVM
- Isolation Forest

Visualization and dimensionality reduction

- Principle Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- T-Distributed Stochastic Neighbor Embedding (t-SNE)

Association Rule Learning

- Apriori
- Eclat

# Semisupervised Learning

Since labeling data is usually time-consuming and costly, you will often have plenty of unlabeled instances, and few labeled instances. Some algorithms can deal with data that's partially labeled. This is called *semisupervised learning*.

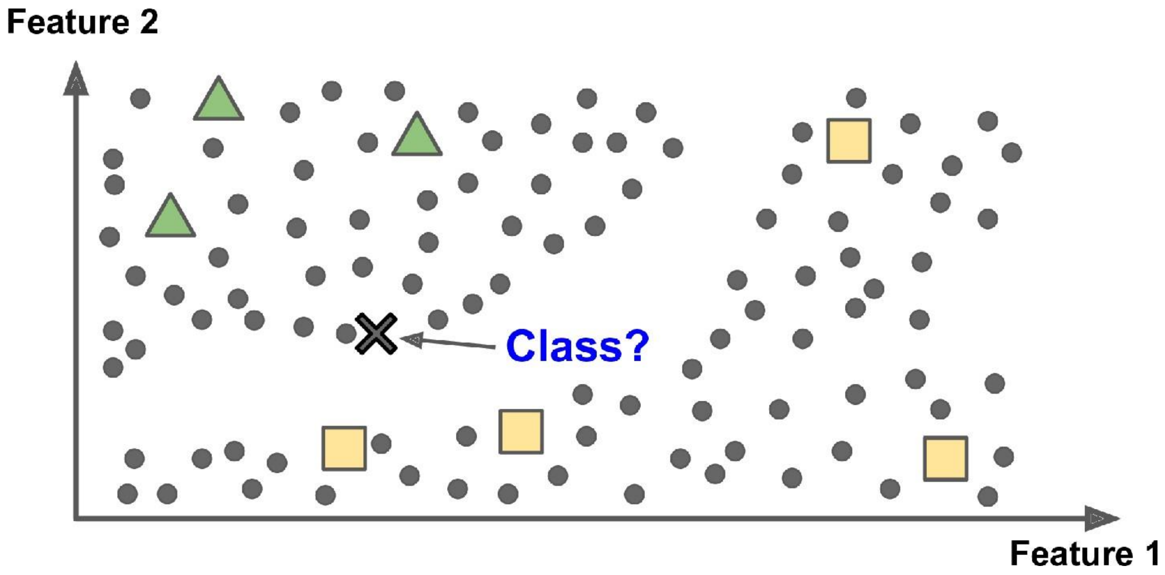
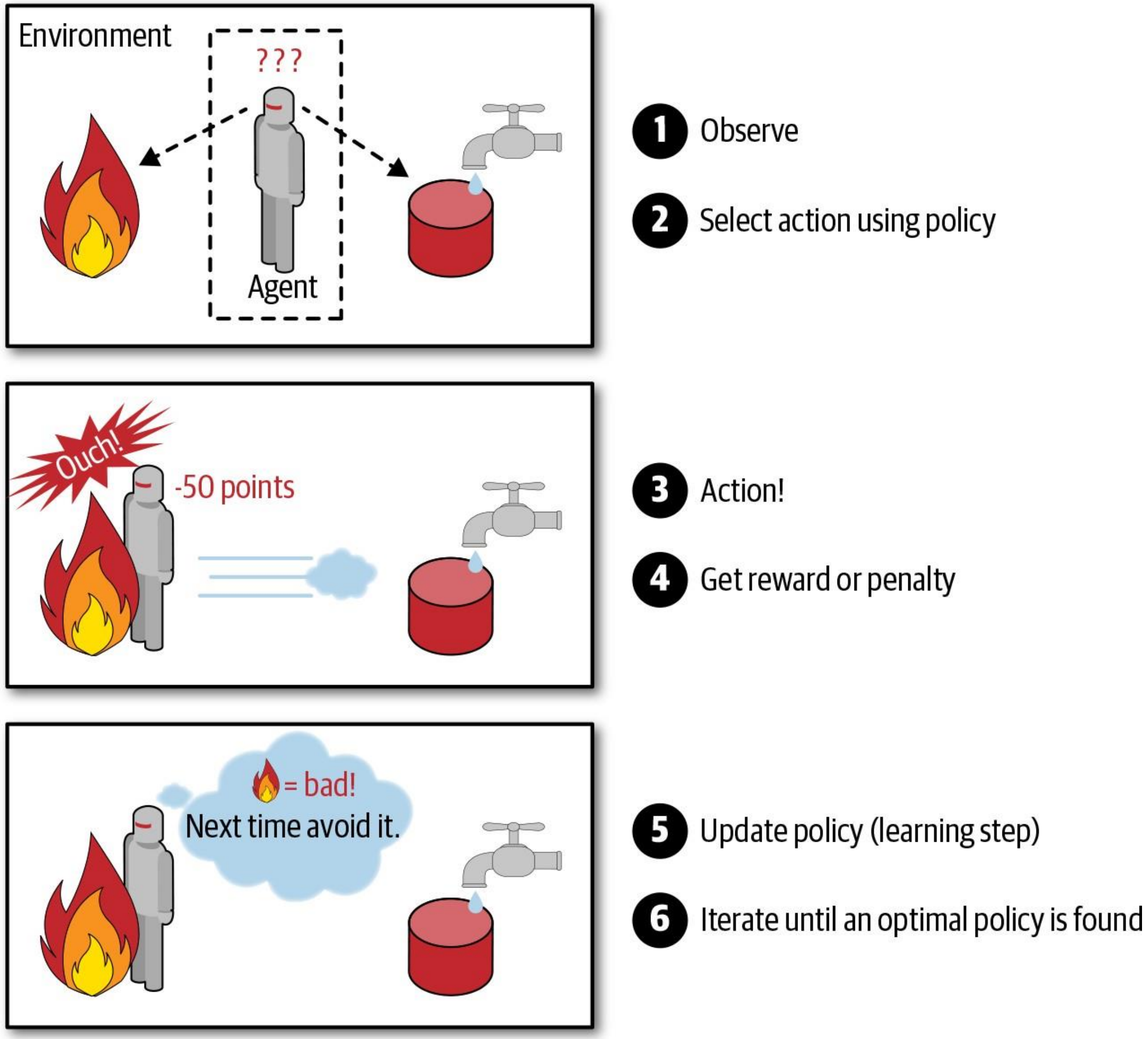


*Figure . Semisupervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares*

Most semisupervised learning algorithms are combinations of unsupervised 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

*Reinforcement Learning* is a very different beast. The learning system, called an *agent* in this context, can observe the environment, select and perform actions, and 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.



1. Observe
2. Select action using policy
3. Action!
4. Get reward or penalty
5. Update policy (learning step)
6. Iterate until an optimal policy is found

For 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 May 2017 when it beat the world champion Ke Jie at the game of Go. 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.

Another criterion used to classify Machine Learning systems is whether or not the system can learn incrementally from a stream of incoming data.

## 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.

**This process needs** a lot of computational power and resources. If your system needs to be able to learn autonomously and it has limited resources (e.g., a smartphone application or a rover on Mars), then carrying around large amounts of training data and taking up a lot of resources to train for hours every day is a showstopper.

**Fortunately, a better option in all these cases is to use algorithms that are capable of learning incrementally.**

# 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.

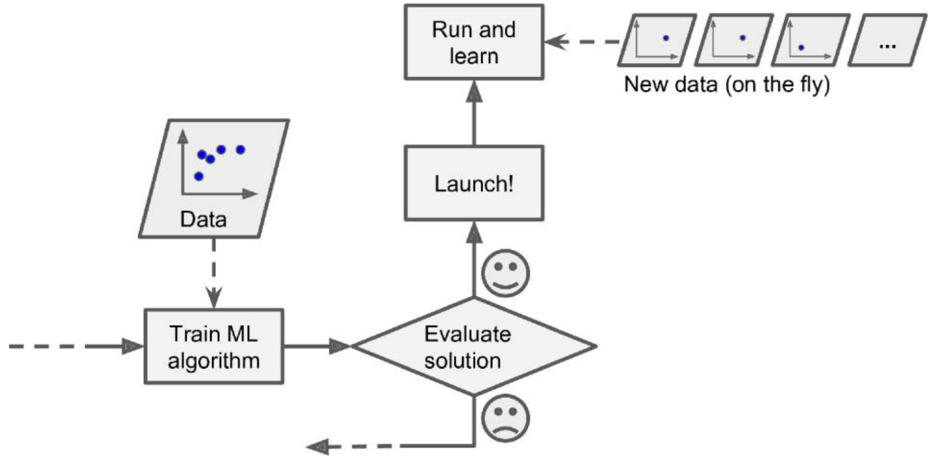


*Figure. In online learning, a model is trained and launched into production, and then it keeps learning as new data comes in*

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.

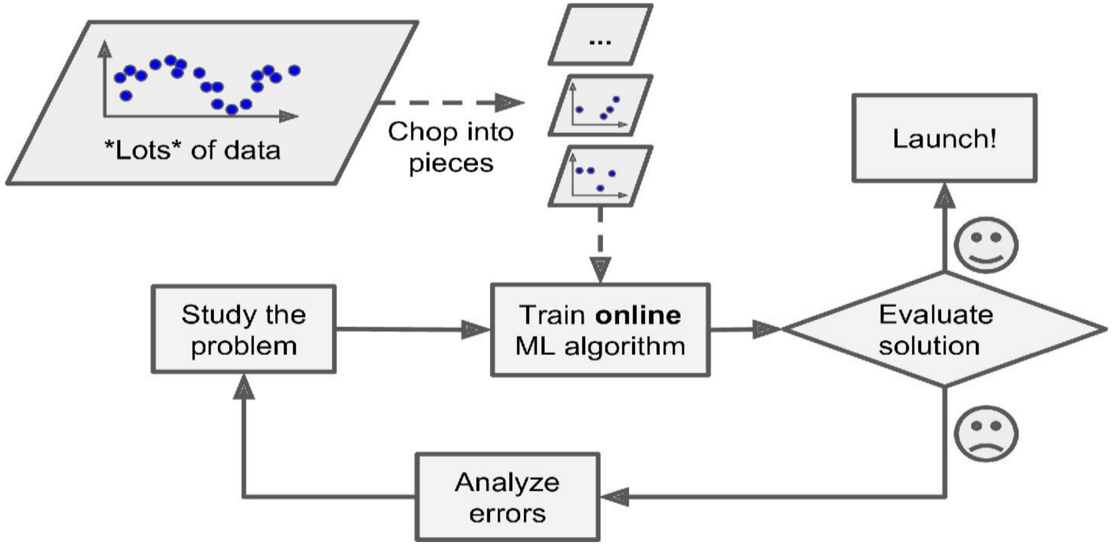


*Figure. Using online learning to handle huge datasets*

**A big challenge** with online learning is that if bad data is fed to the system, the system’s performance will gradually decline. If it’s a live system, your clients will notice. 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 want to monitor the input data and react to abnormal data (e.g., using an anomaly detection algorithm).

## Batch Vs Online Learning

- Training of batch learning is a separate process on the other hand it is a continuous process in online learning.
- Batch learning is best where there is a huge computational power. But online learning can be done in a limited resource.

## Instance-Based Learning

***Instance-based learning*****:** the system learns the examples by heart, then generalizes to new cases by using a similarity measure to compare them to the learned examples (or a subset of them). For example, the new instance would be classified as a triangle because the majority of the most similar instances belong to that class.

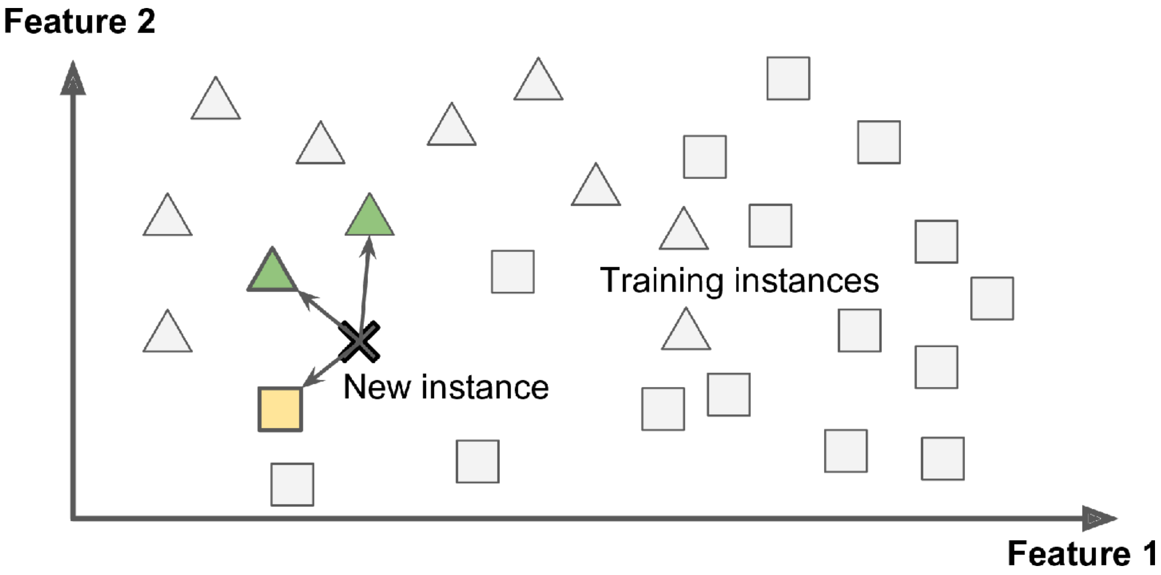


*Figure- Instance-Based Learning*

## 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* .

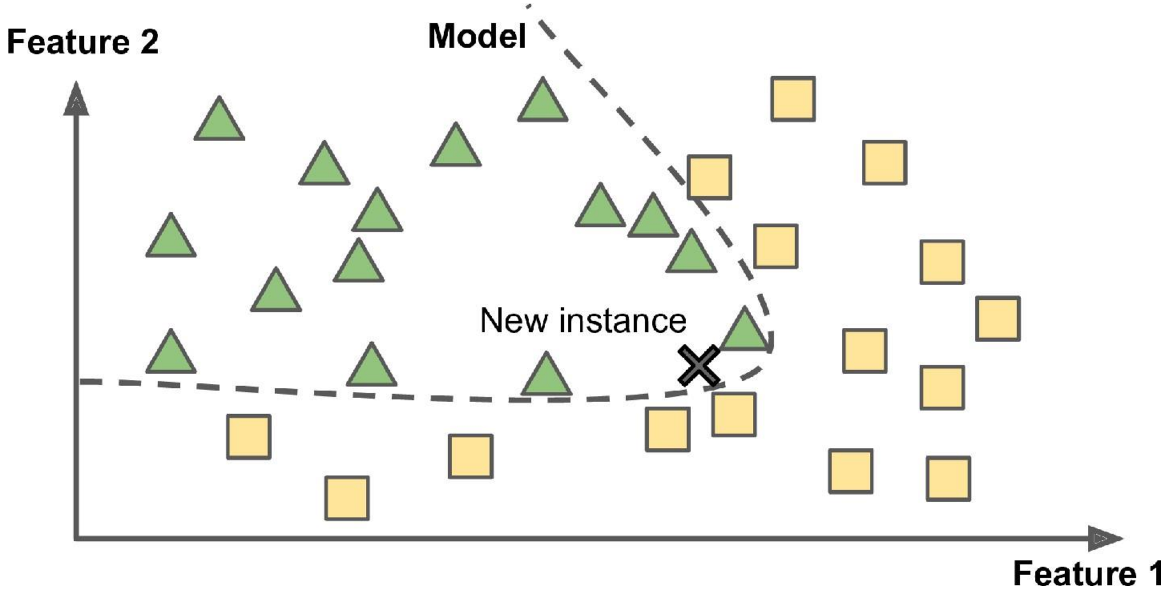


*Figure- Model-based learning*

# Main Challenges of Machine Learning

In short, since your main task is to select a learning algorithm and train it on some data, the two things that can go wrong are “bad algorithm” and “bad data.” Let’s start with examples of bad data.

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Irrelevant Features
- Poor-quality data
- Overfitting the training data

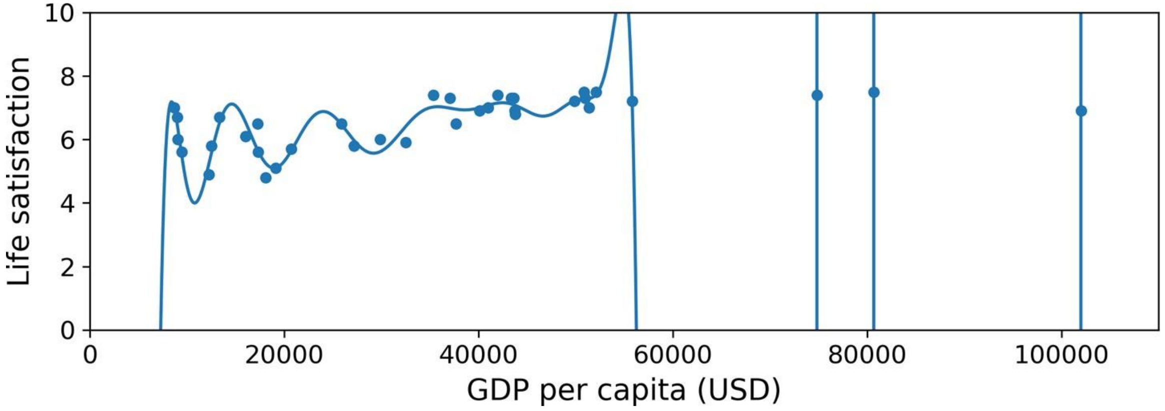


*Figure- Overfitting the training data*

- Underfitting the training data

# Thank you