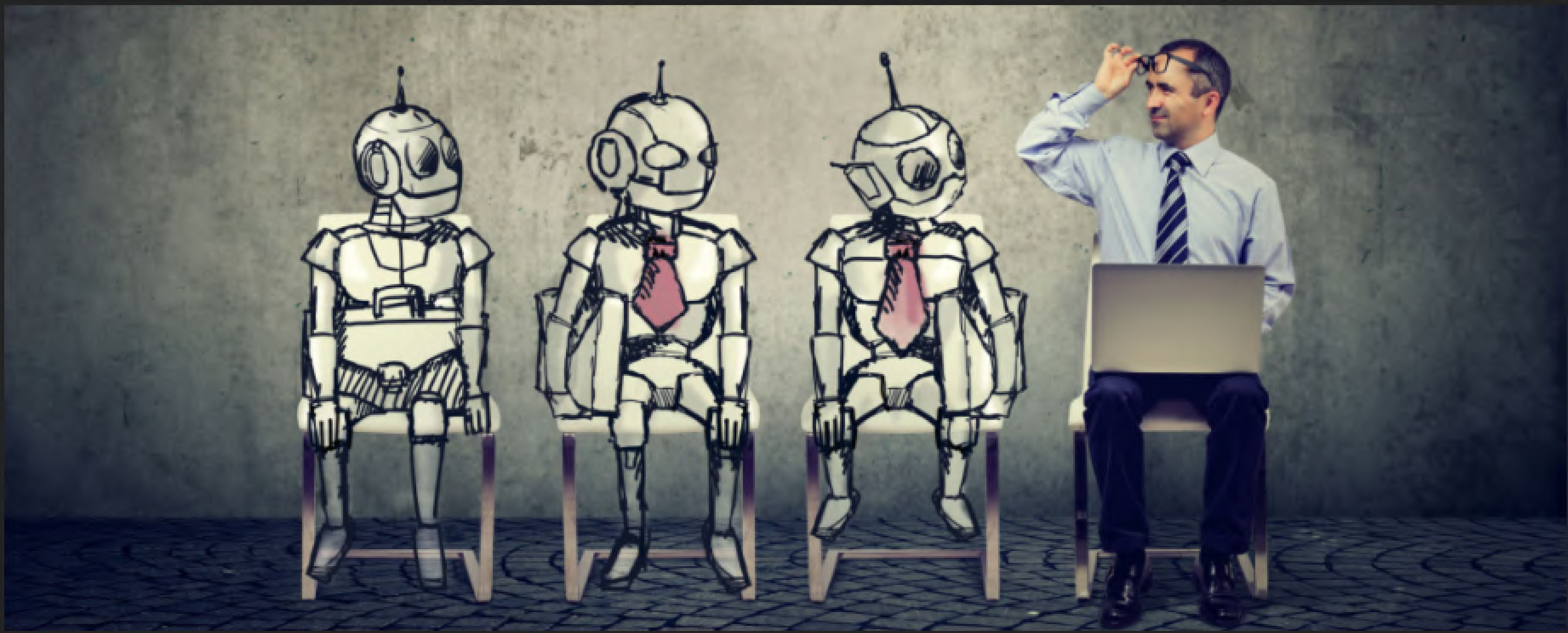
INTRODUCTION TO MACHINE LEARNING

STAY UPDATED FOR MORE TOPICS

TOPICS COVERED:-

1. What is machine learning?
2. Types of machine learning
3. Supervised learning
4. Unsupervised learning
5. Semi-Supervised learning
6. Reinforcement learning
7. Batch learning
8. Online learning
9. Instance based learning
10. Model based learning

WHAT IS MACHINE LEARNING ?



General definition: -
Field of study that gives computer the ability to learn without being explicitly programmed.

Engineering-oriented definition: -
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.

Types of Machine Learning: -

1. There are so many different types of Machine Learning systems that it is useful to classify them in broad categories base on:

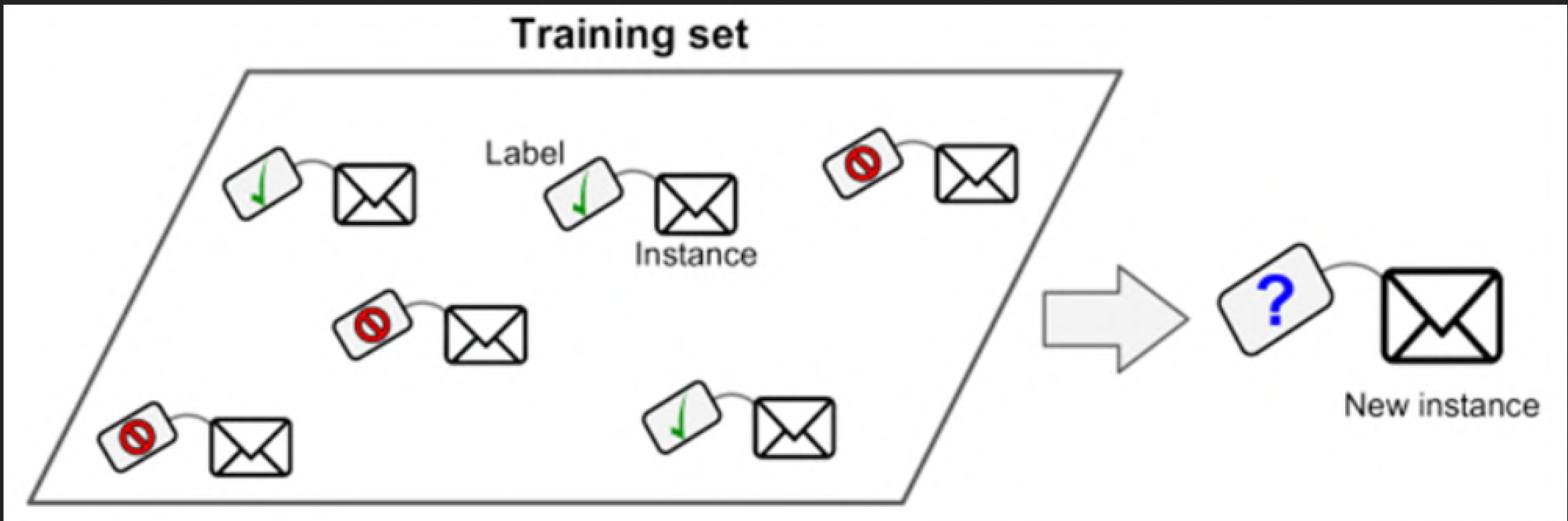
2. Whether or not they are trained with human supervision (supervised, unsupervised, semisupervised and Reinforcement Learning).

3. Whether or not they can learn incrementally (online vs batch learning).

4. Whether or not they work by simply comparing new data points to know data points or instead detect patterns in the training data and build a predictive model (instance based vs model based learning).

Supervised Learning

The training data you feed to the algorithm includes the desired solutions called labels.



A labeled training set for supervised learning (e.g spam classification)

Supervised Learning

Supervised Learning can be further divided into:

Classification: -
The goal is to predict the category or class label of new instances based on past observations.
The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

Regression: -
The goal is to predict continuous numeric values based on input features
Example:- predicting the price of a car, given a set of features (mileage, age, brand, etc) .

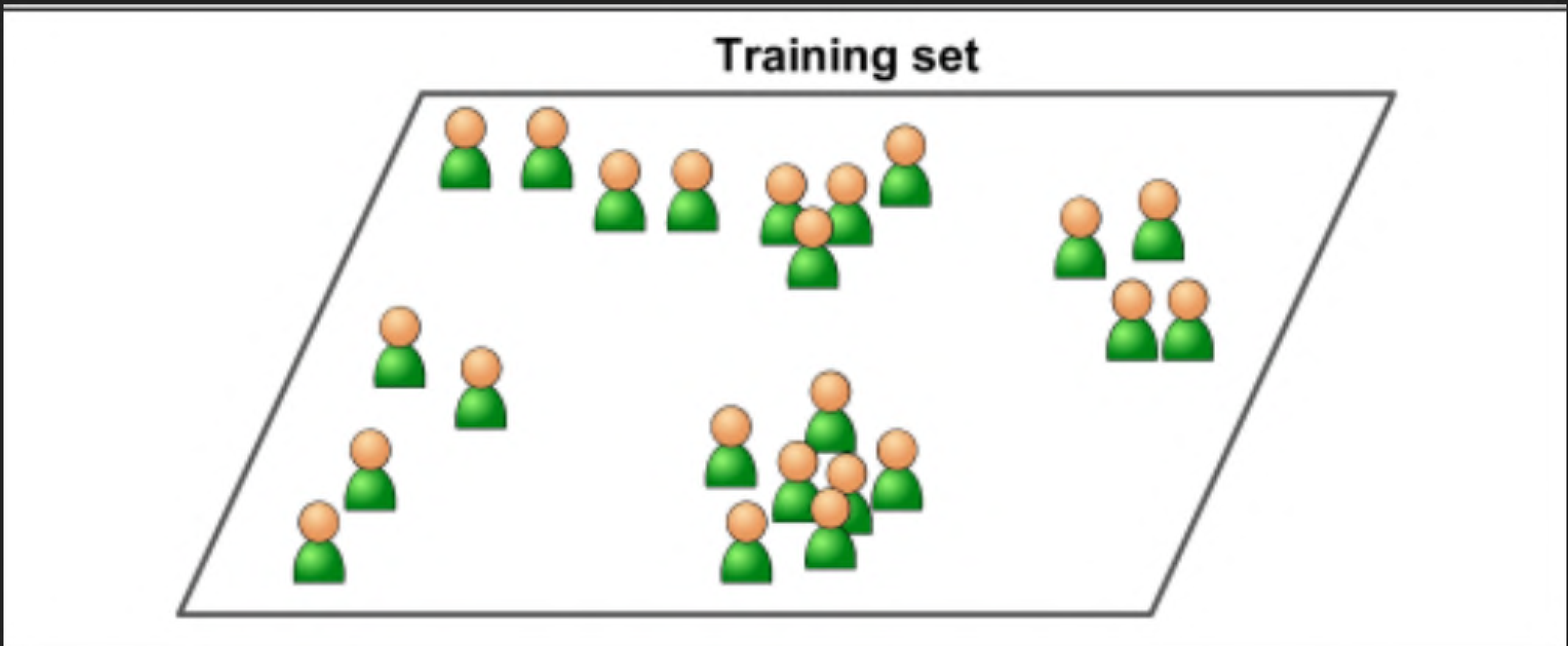
Supervised Learning

Here are some of the most important supervised learning algorithms:

1. k-Nearest Neighbors
2. Linear Regression
3. Logistic Regression
4. Support Vector Machines (SVMs)
5. Decision Trees and Random Forests
6. Neural netowrks

Unsupervised Learning

As you might guess, the training data is unlabeled. The system learns without a teacher.



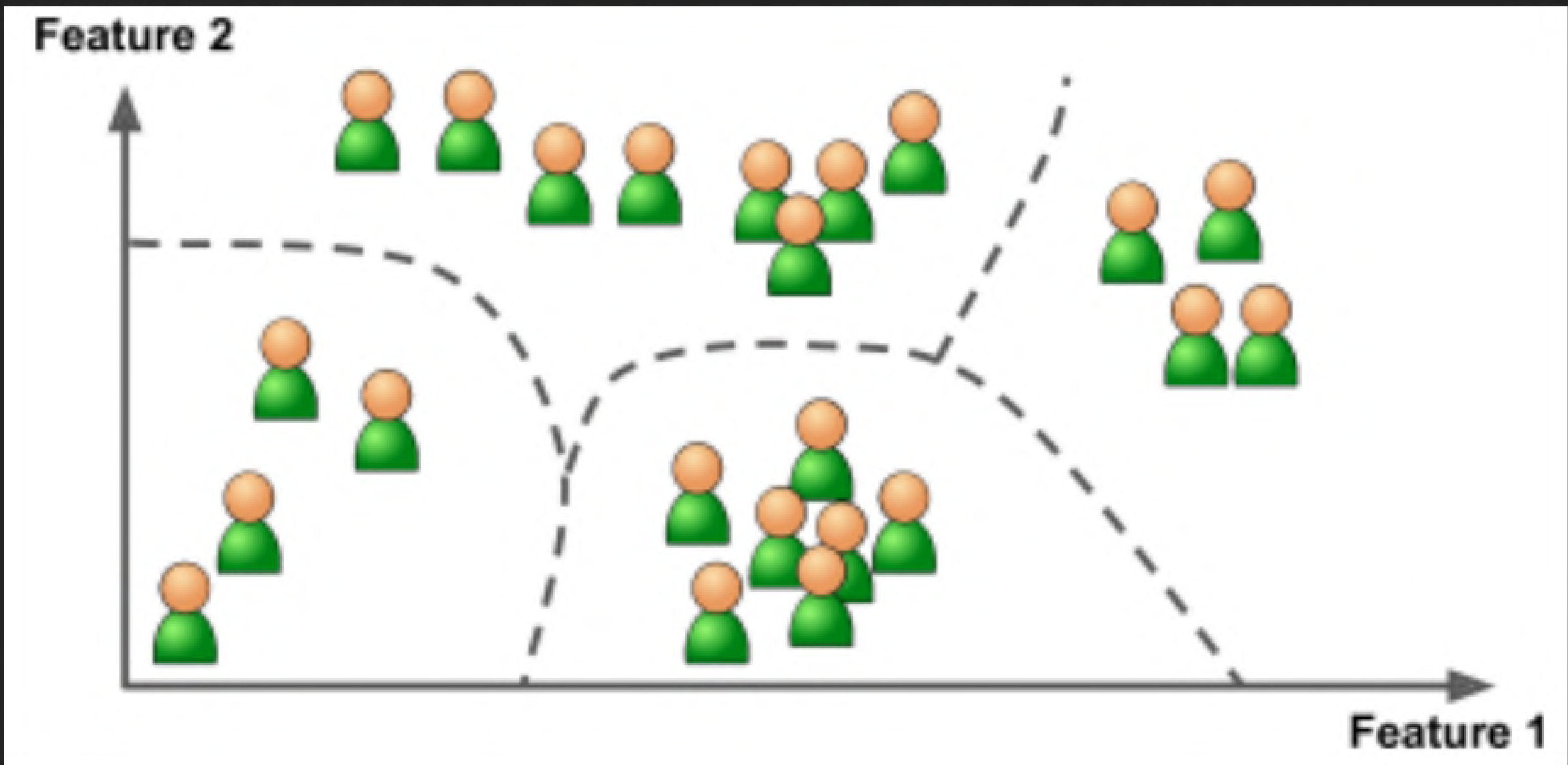
An unlabeled training set for unsupervised learning.

Unsupervised Learning

Unsupervised learning is further divided into:

I. Clustering

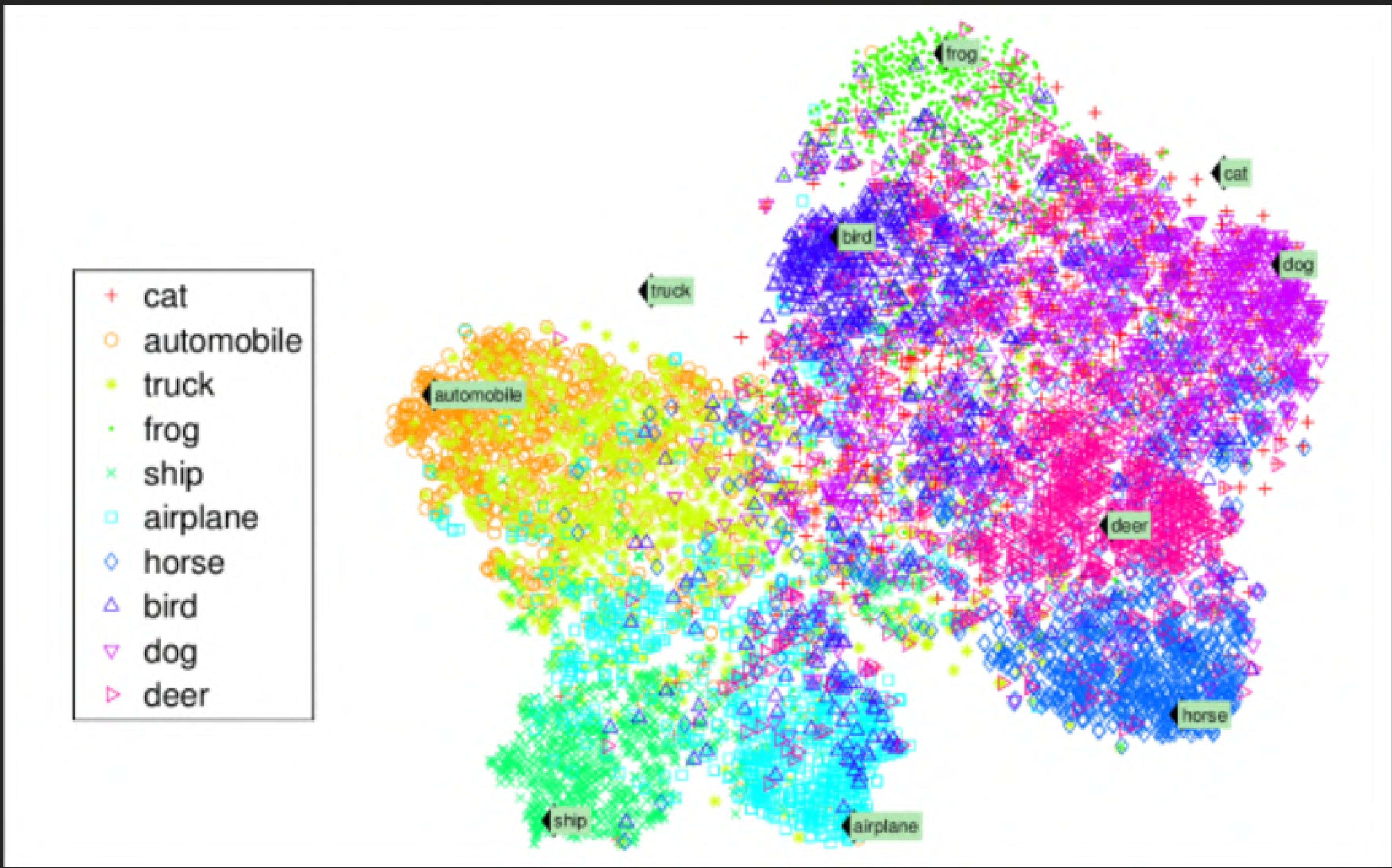
Say you have a lot of data about your blog's visitors. You may want to run a clustering algorithm to try to detect groups of similar visitors.
At no point do you tell the algorithm which group a visitor belongs to: it finds those connections without your help.
For example, it might notice that 40% of your visitors are males who love comic books and generally read your blog in the evening, while 20% are young sci-fi lovers who visit during the weekends, and so on.



Unsupervised Learning

2. Visualization Algorithms

Visualization algorithms are also good examples of unsupervised learning algorithms: you feed them a lot of complex and unlabeled data, and they output a 2D or 3D rep-resentation of your data that can easily be plotted.
These algorithms try to preserve as much structure as they can (e.g., trying to keep separate clusters in the input space from overlapping in the visualization), so you can understand how the data is organized and perhaps identify unsuspected patterns



Unsupervised Learning

3. Dimensionality Reduction

The goal is to simplify the data without losing too much information.
One way to do this is to merge several correlated features into one. For example, a car's mileage may be very correlated with its age, so the dimensionality reduction algorithm will merge them into one feature that represents the car's wear and tear. This is called feature extraction.

4. Association rule learning

The goal is to dig into large amounts of data and discover interesting relations between attributes.
For example, suppose you own a supermarket. Running an association rule on your sales logs may reveal that people who purchase barbecue sauce and potato chips also tend to buy steak. Thus, you may want to place these items close to each other.

Unsupervised Learning

5. Anomaly Detection

Yet another important unsupervised task is anomaly detection for example, detecting unusual credit card transactions to prevent fraud, catching manufacturing defects, or automatically removing outliers from a dataset before feeding it to another learning algorithm.

The system is shown mostly normal instances during training, so it learns to recognize them and when it sees a new instance it can tell whether it lookslike a normal one or whether it is likely an anomaly.

A very similar task is novelty detection: the difference is that novelty detection algorithms expect to see only normal data during training, while anomaly detection algorithms are usually more tolerant, they can often perform well even with a small percentage of outliers in the training set.

Unsupervised Learning

Here are some of the most important unsupervised learning algorithms:

1. Clustering
K-Means
DBSCAN
Hierarchical Cluster Analysis (HCA)

2. Anomaly detection and novelty detection
One-class SVM
Isolation Forest

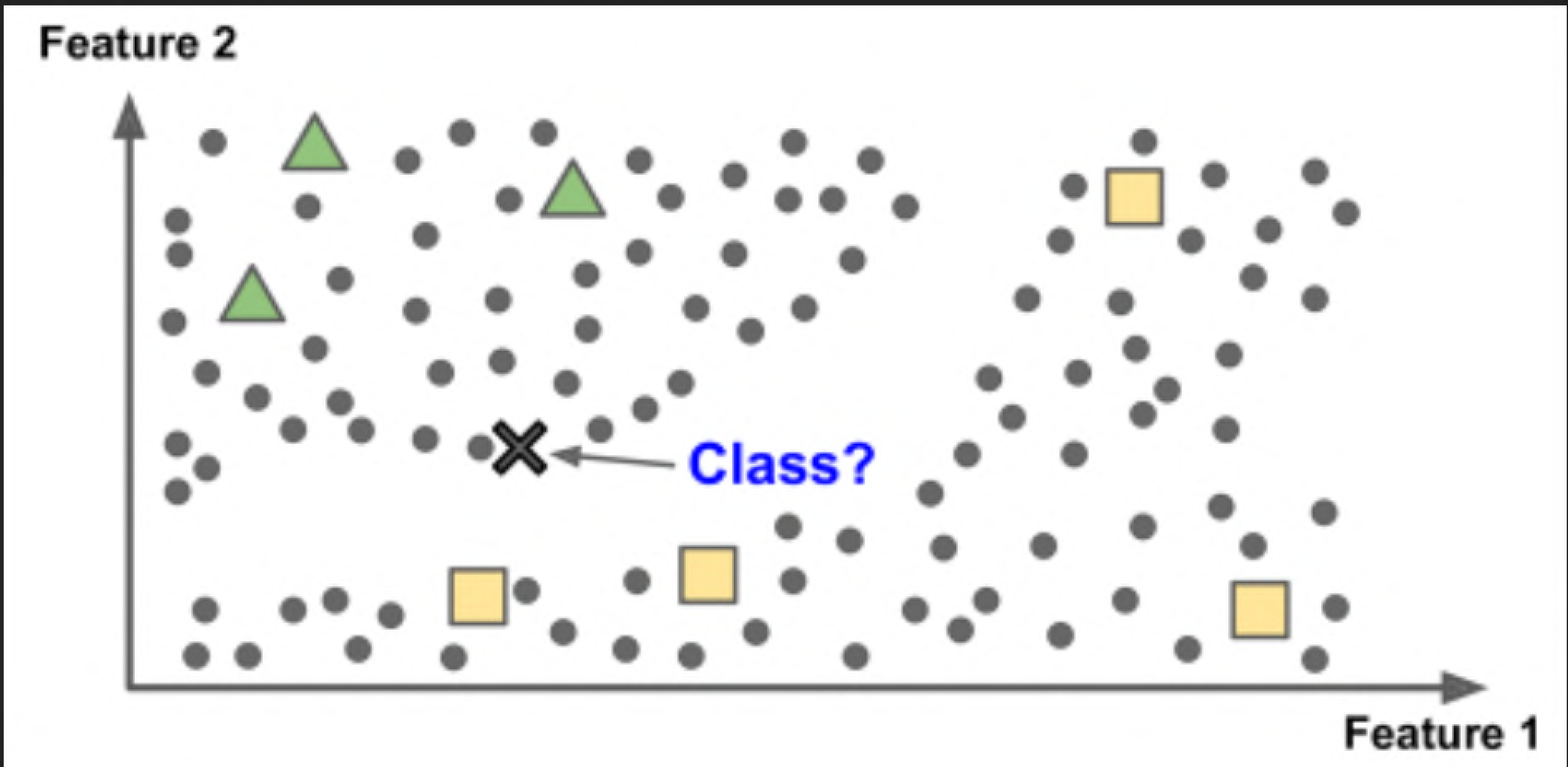
3. Visualization and dimensionality reduction
Principal Component Analysis (PCA)
Kernel PCA
Locally-Linear Embedding (LLE)
t-distributed Stochastic Neighbor Embedding (t-SNE)

4. Association rule learning
Apriori
Eclat

Semi-supervised Learning

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called semisupervised learning.

Google Photos, are good examples of this.
Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in photos 2, 5, and 7. This is the unsupervised part of the algorithm (clustering)..



Semi-supervised Learning

Most semisupervised learning algorithms are combinations of unsupervised and supervised algorithms.

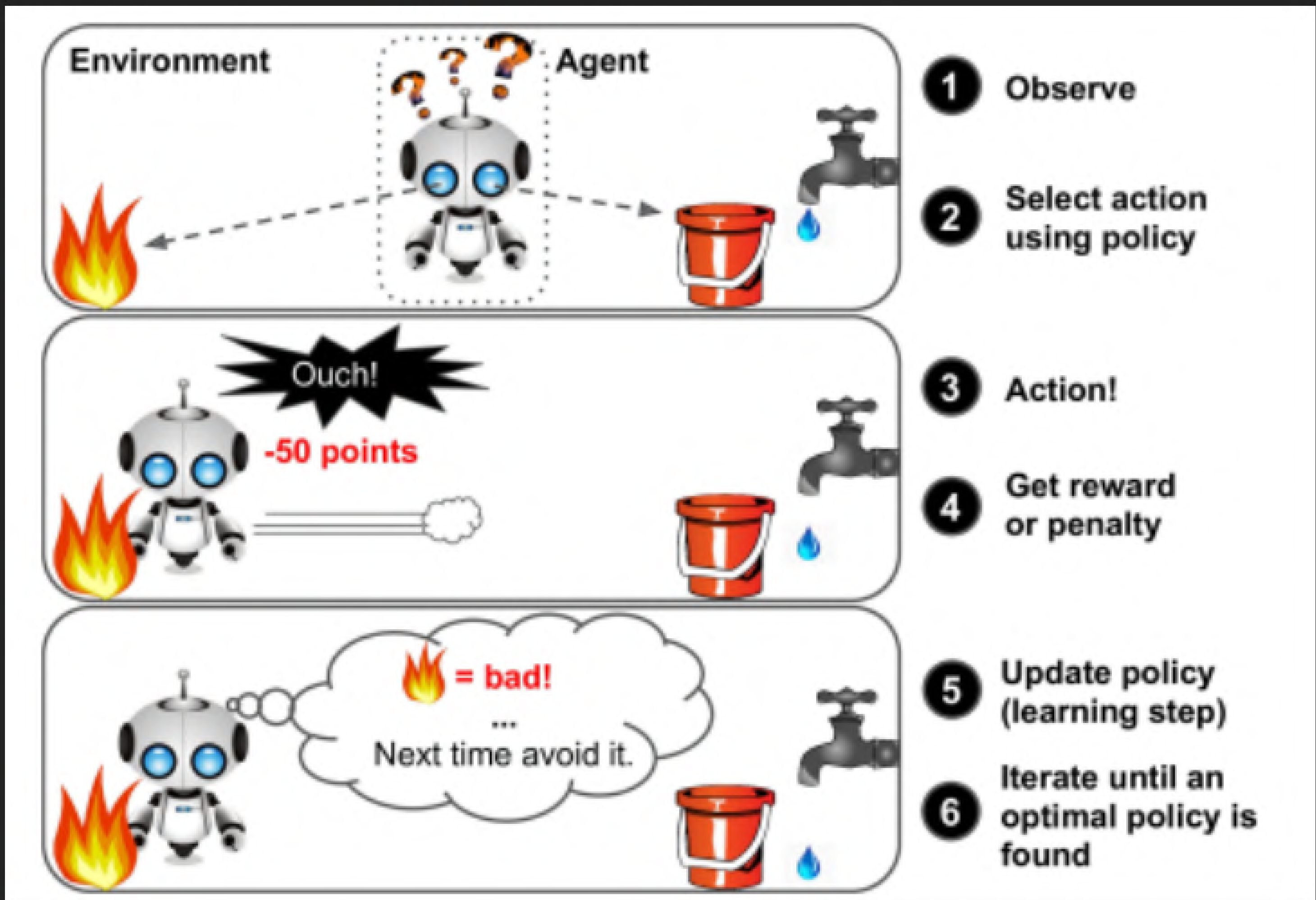
For example, deep belief networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on top of one another.

Reinforcement Learning

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.



Reinforcement Learning

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.

Batch Learning

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

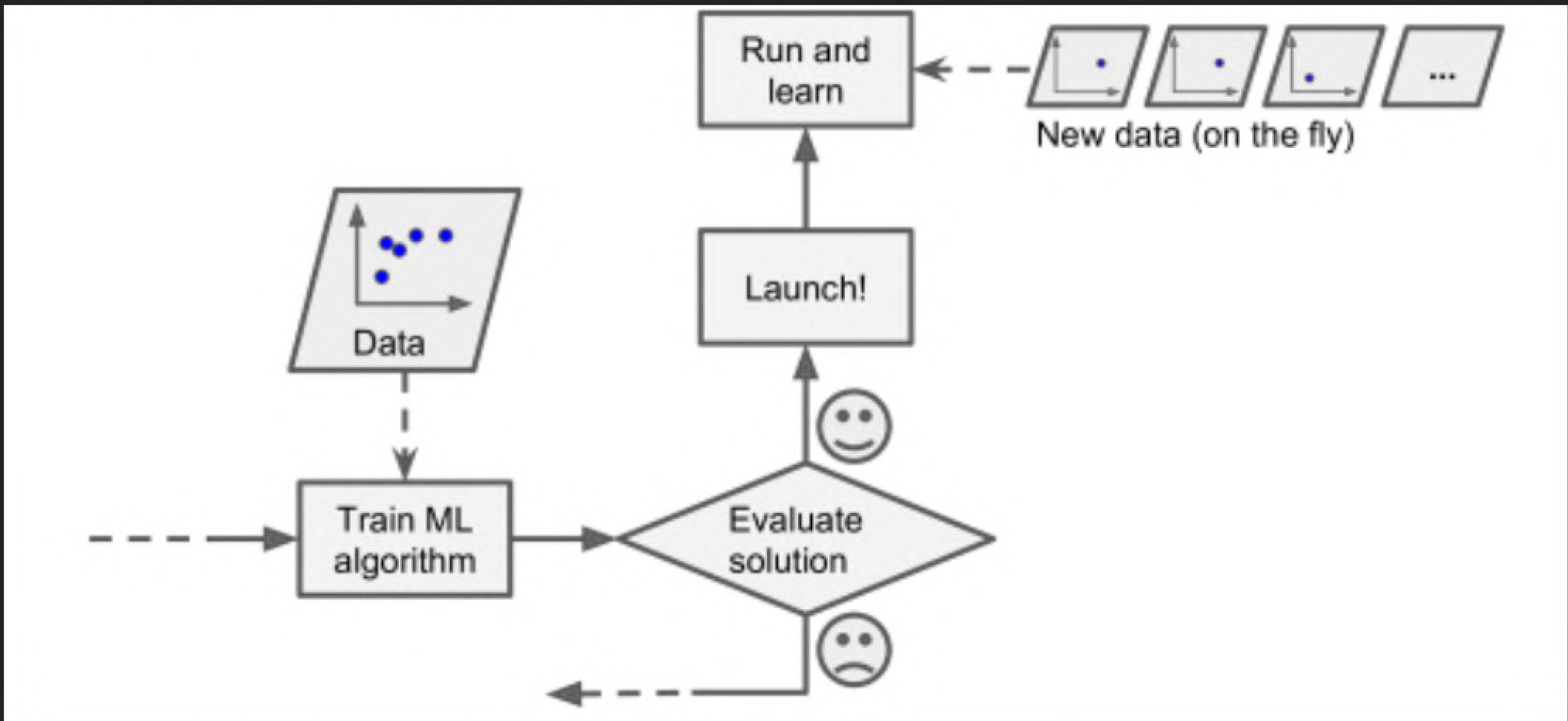
This will generally take a lot of time and computing resources, so it is typically done offline.

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.

Online Learning

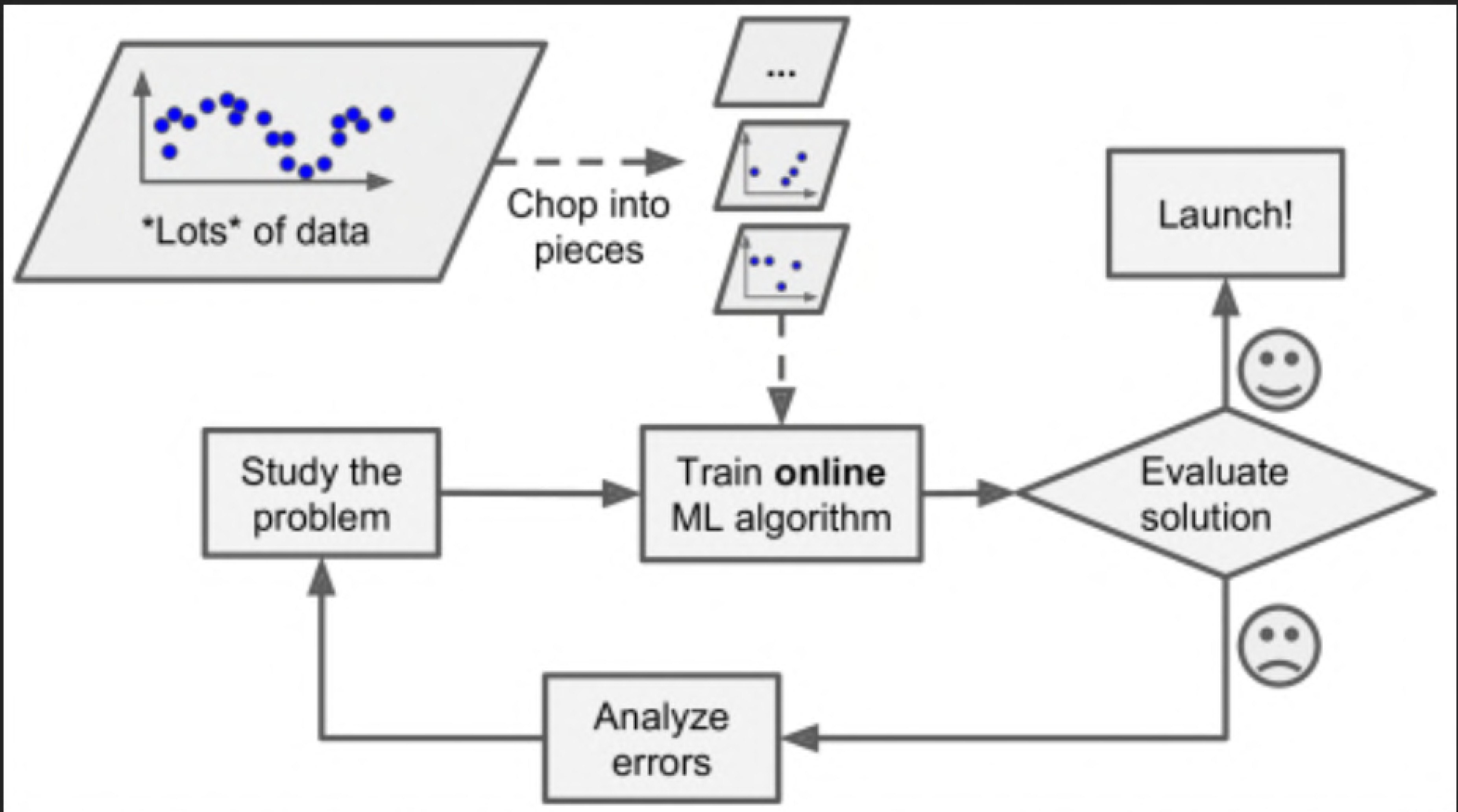
In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrive.



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.

Online Learning

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.



Out-of-core learning is usually done offline (i.e., not on the live system), so online learning can be a confusing name. Think of it as incremental learning.

Online Learning

One important parameter of online learning systems is how fast they should 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 points (outliers).
A big challenge with online learning is that if bad data is fed to the system, the system's performance will gradually decline.
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)

Instance Based Learning

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

Instance Based Learning

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

Model Based Learning

Another way to generalize from a set of examples is to build a model of these examples, then use that model to make predictions. This is called model-based learning.