1

ML - Kick off

In this age of modern technology, there is one resource that we have in abundance: a large amount of structured and unstructured data.

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

Engineering-oriented ML 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.

Tom Mitchell, 1997

Spam filter example

The task T is to flag spam for new emails, the experience E is the training data, and the performance measure P could be the ratio of correctly classified emails.

Machine learning is great for:

- Problems for which existing solutions require a lot of hand-tuning or long lists of rules.
- Complex problems for which there is no good solution at all using a traditional approach
- Fluctuating environments
- Getting insights about complex problems and large amounts of data. (data mining)

The three different types of machine learning

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

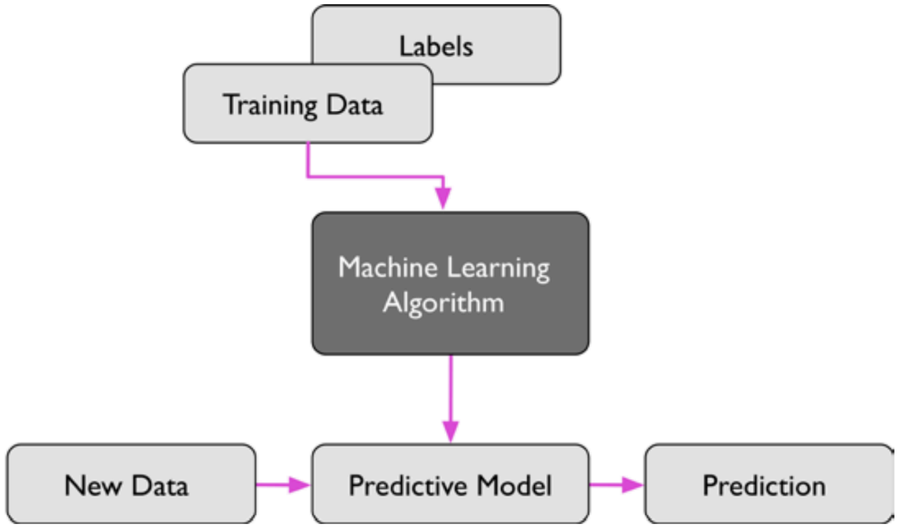
- No labels/targets
- No feedback
- Find hidden structure in data

Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

The term supervised refers to a set of samples where the desired output signals (labels) are already known.

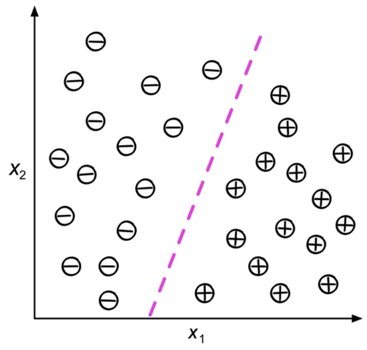
Supervised Learning



Classification

A supervised learning task with categorical class labels.

Those class labels are discrete, unordered values that can be understood as the group memberships of the instances.

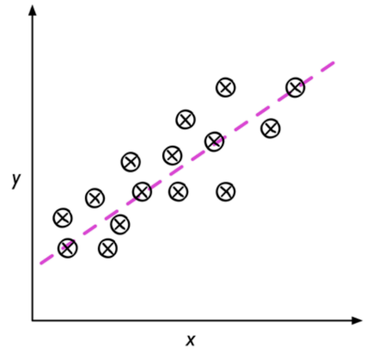


Regression

The outcome is a continuous value.

We have predictors: (explanatory) variables and a continuous response variable(s) (outcome or target).

The goal is to find a relationship between those variables that allows us to predict an outcome.



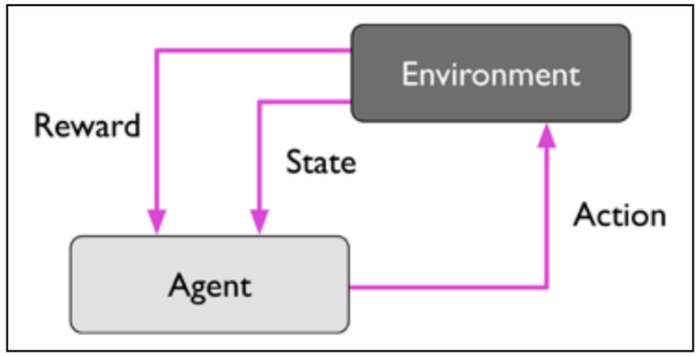
Regression note

Fun fact: this odd-sounding name is a statistics term introduced by Francis Galton while he was studying the fact that the children of tall people tend to be shorter than their parents. Since children were shorter, he called this *regression to the mean*. This name was then applied to the methods he used to analyze correlations between variables.

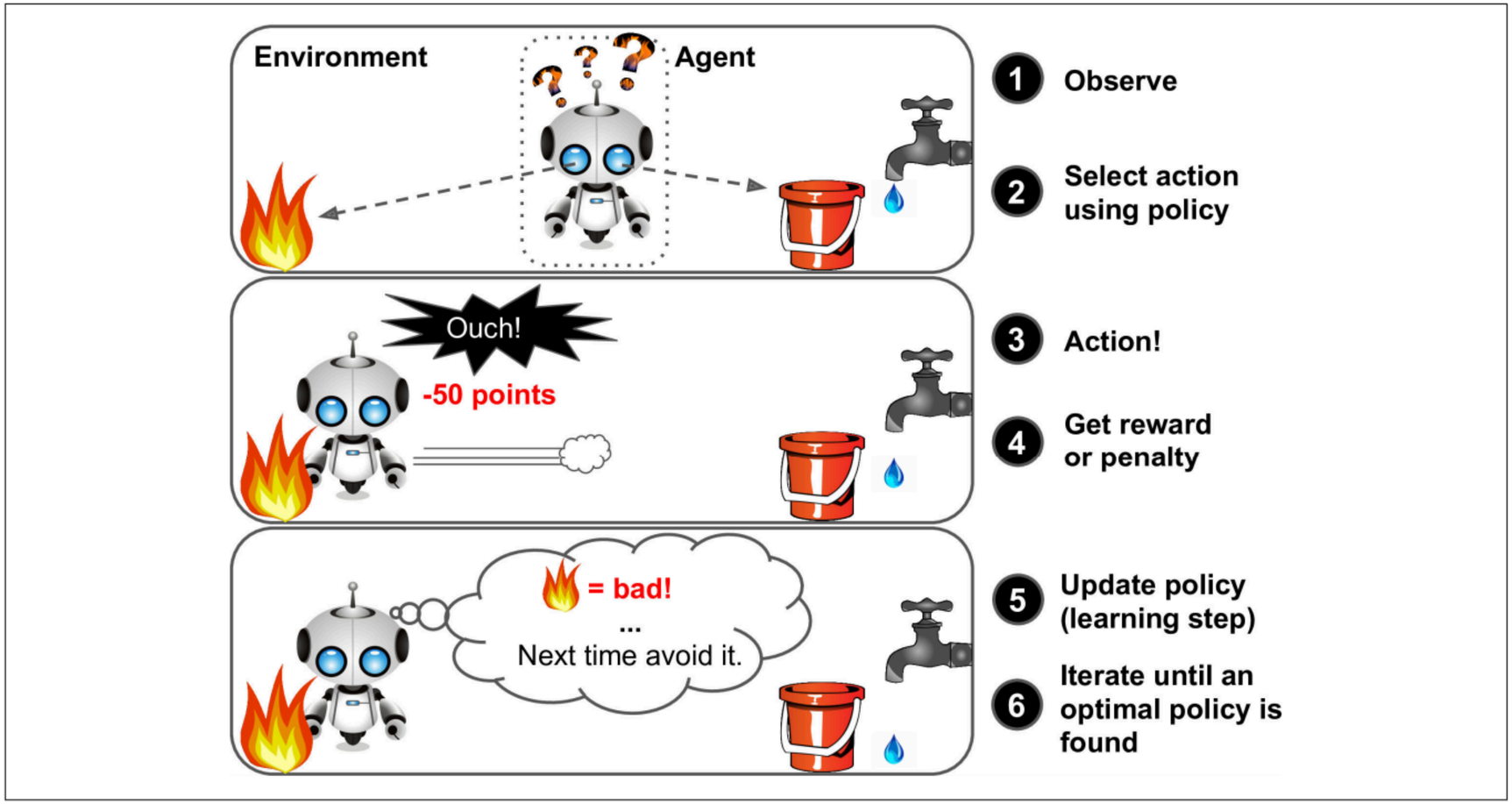
The goal is to develop a system (agent) that improves its performance based on interactions with the environment.

Reinforcement Learning

- The information about the current state of the environment typically also includes a so-called reward signal
- This feedback is not the correct ground truth label or value, but a measure of how well the action was measured by a reward function.
- An agent can learn a series of actions that maximizes this reward via an exploratory trial-and-error approach or deliberative planning.



Reinforcement Learning Flow



Reinforcement Learning: Chess example

The outcome of each move can be thought of as a different state.

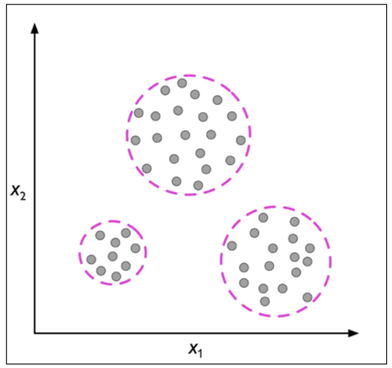
- Visiting certain locations on the chess board as being associated with a positive event.
- Other positions are associated with a negative event, such as losing a chess piece to the opponent in the following turn.
- Reinforcement learning is also concerned with learning the series of steps by maximizing a reward based on immediate and delayed feedback.

To explore the structure of our data to extract meaningful information without the guidance

Clustering

To organize a pile of information into meaningful subgroups (clusters).

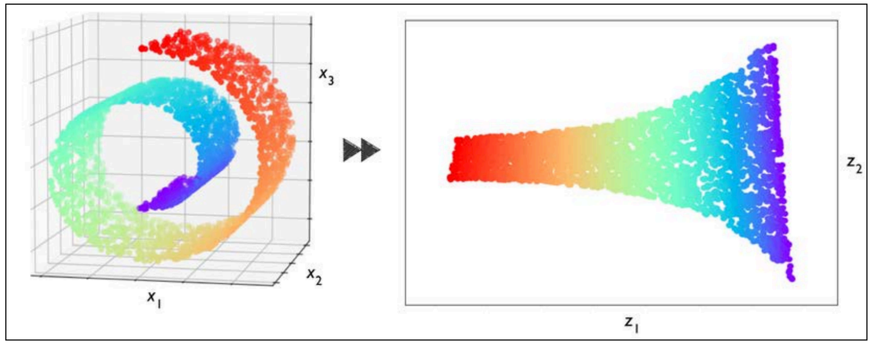
Each cluster that arises during the analysis defines a group of objects that share a certain degree of similarity but are more dissimilar to objects in other clusters



Dimensionality Reduction

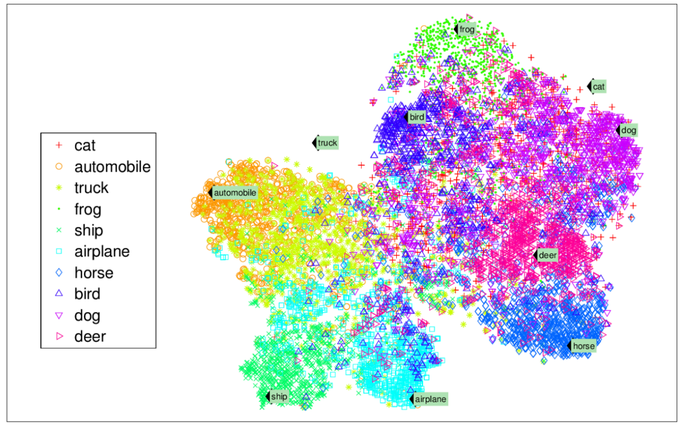
Feature preprocessing to remove noise from data.

Compress the data onto a smaller dimensional subspace while retaining most of the relevant information.



Visualization

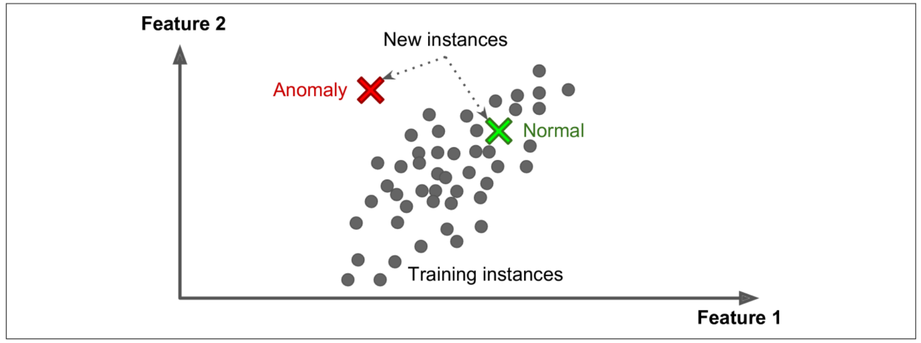
These algorithms try to preserve as much structure as they can (e.g., trying to keep separate clusters in the input space from overlapping in the visualization)



Anomaly detection

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 looks like a normal one or whether it is likely an anomaly.

A very similar task is novelty detection.



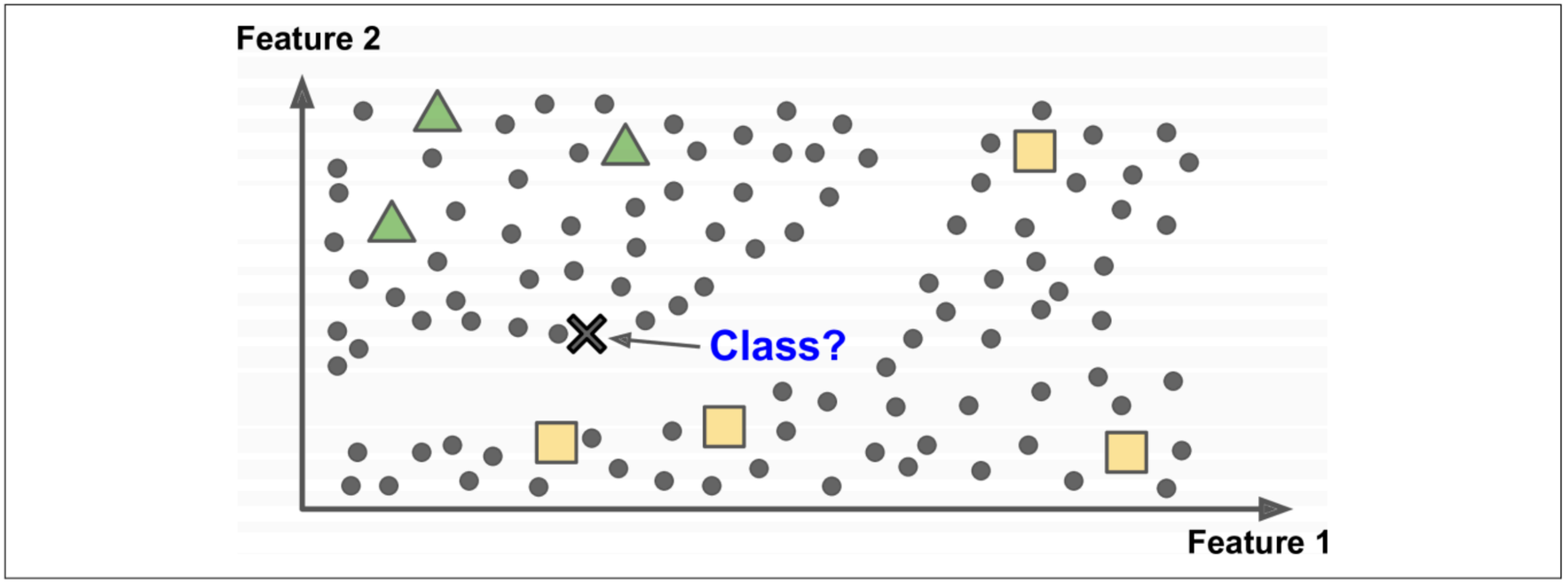
Association rule learning

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.

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data.

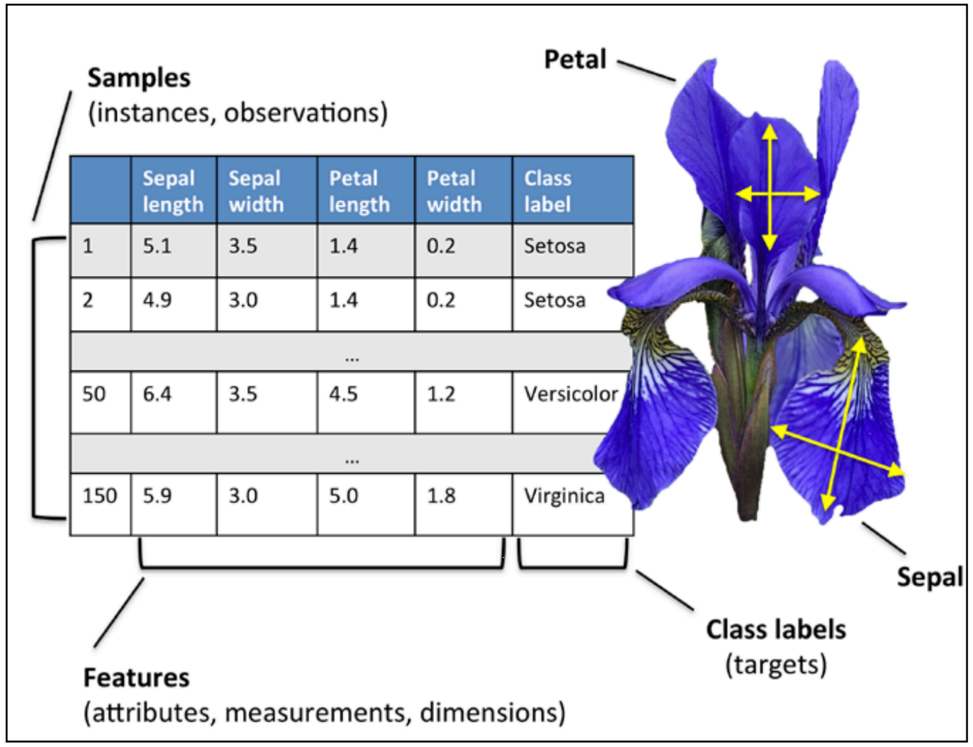
Semisupervised Learning



Basic terminology and notations

Let's use as example the Iris Dataset.

Here, each flower sample represents one row in our dataset



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

$$\boldsymbol{X} \in \mathbb{R}^{150\times 4}$$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Basic terminology and notations

We will use the superscript i to refer to the ith training sample, and the subscript j to refer to the jth dimension of the training dataset.

Lowercase bold-face letters to refer to vectors.

Uppercase, bold-face letters to refer to matrices.

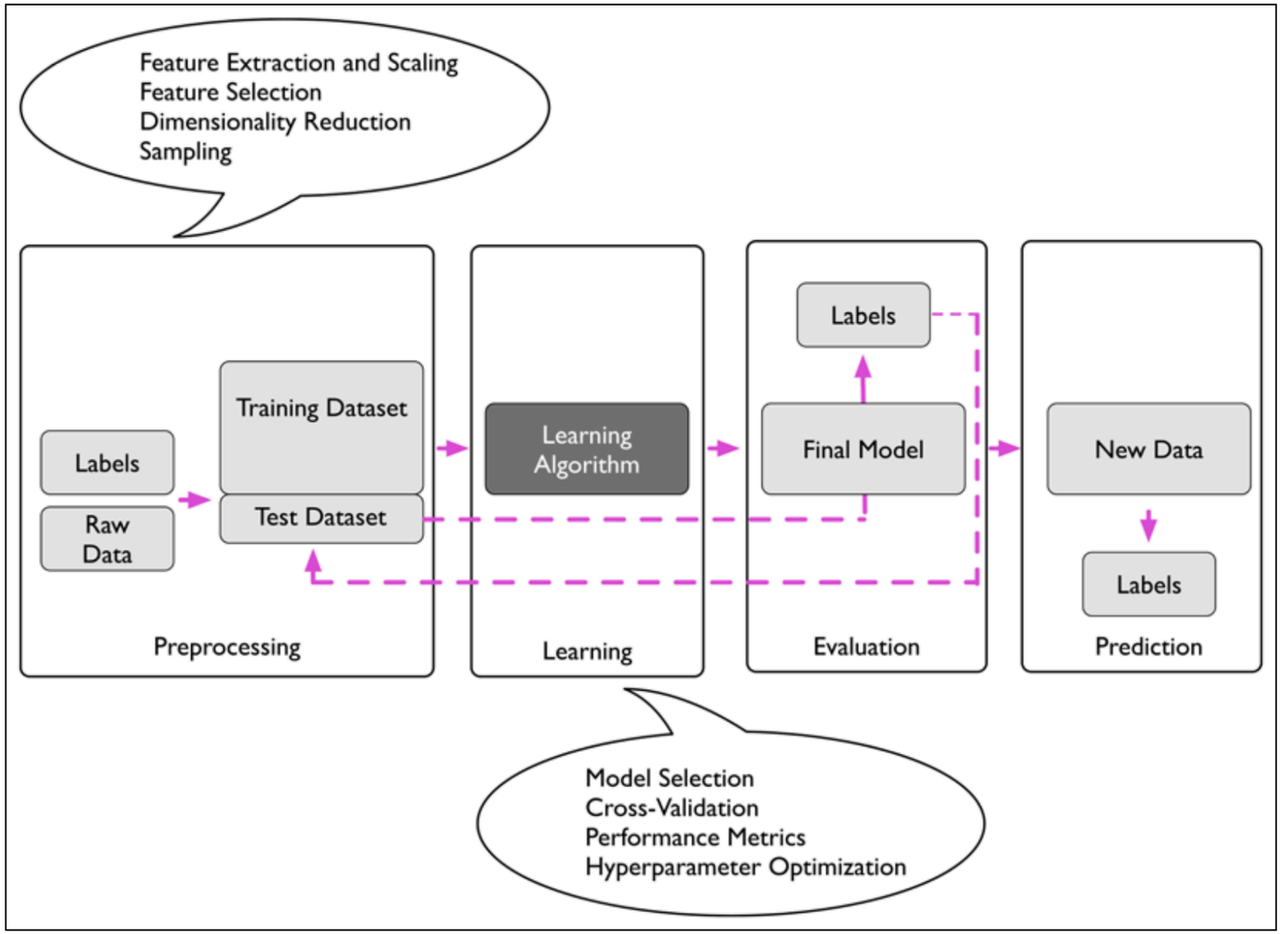
$$\boldsymbol{X} \in \mathbb{R}^{150 \times 4}$$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

$$\boldsymbol{x}_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(150)} \end{bmatrix}$$

Roadmap for ML systems



Preprocessing

- Raw data rarely comes in the form and shape that is necessary for the optimal performance of a learning algorithm.
- Many machine learning algorithms also require that the selected features are on the same scale for optimal performance.
- Some of the selected features may be highly correlated and therefore redundant to a certain degree.
- Data splitting: we use the training set to train and optimize our machine learning model, while we keep the test set until the very end to evaluate the final model.

Training and selecting a predictive model

- Each algorithm has its inherent biases, and no single model enjoys superiority if we don't make any assumptions about the task.
- We first have to decide upon a metric to measure performance.
- Cross-validation techniques in order to estimate the generalization performance of the model.
- Hyperparameter optimization techniques that help us to fine-tune the performance of our model

Evaluating models and predicting unseen data instances

- If we are satisfied with its performance, we can now use this model to predict new, future data.
- Parameters for the previously mentioned procedures, such as feature scaling and dimensionality reduction, are solely obtained from the training dataset, and the same parameters are later reapplied to transform the test dataset.

Types of Machine Learning Systems depending on perspective

Trained with or without human supervision

- Supervised
- Unsupervised
- Semisupervised
- Reinforcement Learning

Can or cannot learn incrementally the fly

- Online learning
- Batch learning

How they generalize?

- Instance-based learning
- Model-based learning

Batch learning

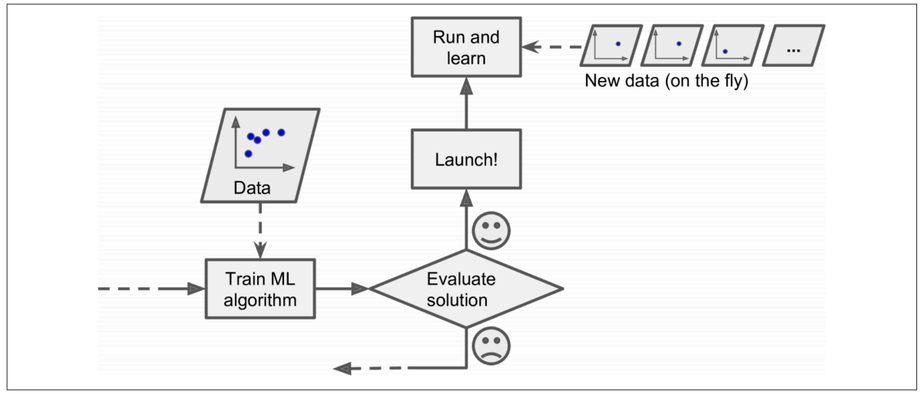
The system is incapable of learning incrementally: it must be trained using all the available data.

Offline learning: The system is trained, and then it is launched into production and runs without learning anymore.

When you have 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).

Online learning

You train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches.



Batch learning

Training on the full set of data requires a lot of computing resources. It will end up costing you a lot of money.

If the amount of data is huge, it may even be impossible to use a batch learning algorithm

Online learning

Need to adapt to change rapidly or autonomously.

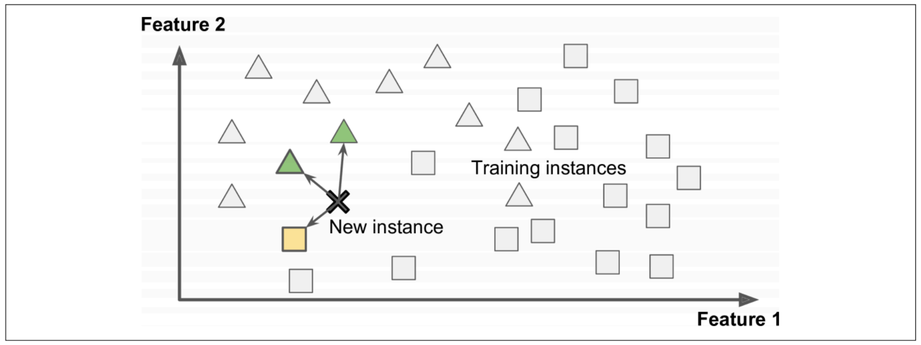
Huge datasets that cannot fit in one machine's main memory.

How fast they should adapt to changing data?

If bad data is fed to the system, the system's performance will gradually decline?

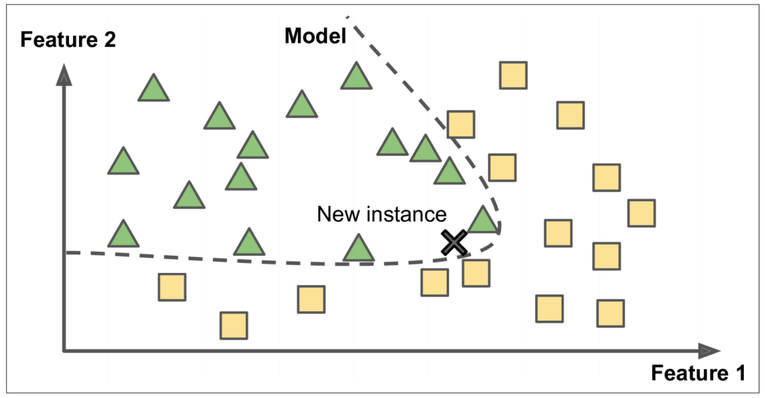
Instance-Based

The system learns the examples by heart, then generalizes to new cases by comparing them to the learned examples, using a similarity measure.



Model-Based

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



Model-Based

- Model selection is required.
- Define model parameters, specify a performance measure (a utility function (or fitness function) that measures how good your model is, or you can define a cost function that measures how bad it is).
- Then, you input the training examples and it finds the parameters that make the model fit best to your data.
- Finally, you applied the model to make predictions on new cases (this is called *inference*), hoping that this model will generalize well.

Insufficient Quantity of Training Data

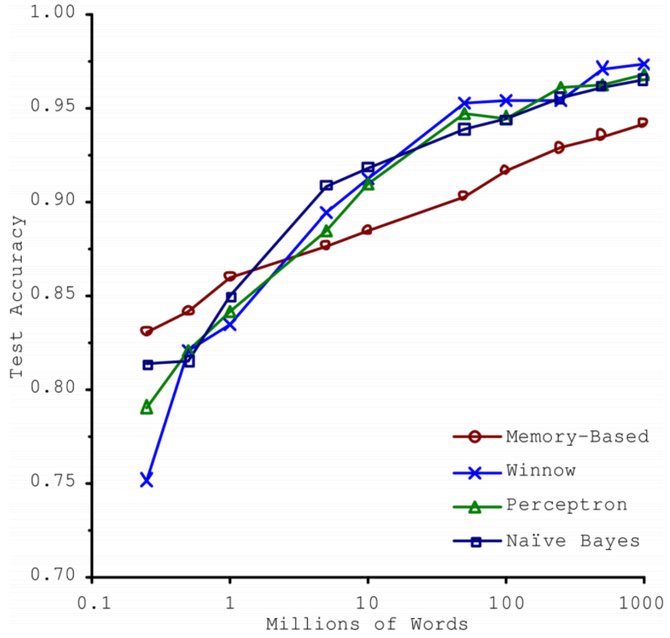
- It takes a lot of data for most Machine Learning algorithms to work properly. Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples.

Main Challenges of Machine Learning

The idea that data matters more than algorithms for complex problems was further popularized by Peter Norvig.

The Unreasonable Effectiveness of Data

In a famous paper published in 2001, Microsoft researchers Michele Banko and Eric Brill showed that very different Machine Learning algorithms, including fairly simple ones, performed almost identically well on a complex problem of natural language disambiguation[8] once they were given enough data (as you can see in Figure 1-20).



Nonrepresentative Training Data

- In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to.
- This is often harder than it sounds: if the sample is too small, you will have sampling noise (i.e., nonrepresentative data as a result of chance).
- Very large samples can be nonrepresentative if the sampling method is flawed (sampling bias).

Poor-Quality Data

- It is often well worth the effort to spend time cleaning up your training data. The truth is, most data scientists spend a significant part of their time doing just that.
- If some instances are clearly outliers, it may help to simply discard them or try to fix the errors manually.
- If some instances are missing a few features (e.g., 5% of your customers did not specify their age), you must decide what to do.

Irrelevant Features

- A critical part of the success of a Machine Learning project is coming up with a good set of features to train on.

Feature Engineering

- Feature selection: selecting the most useful features to train on among existing features.

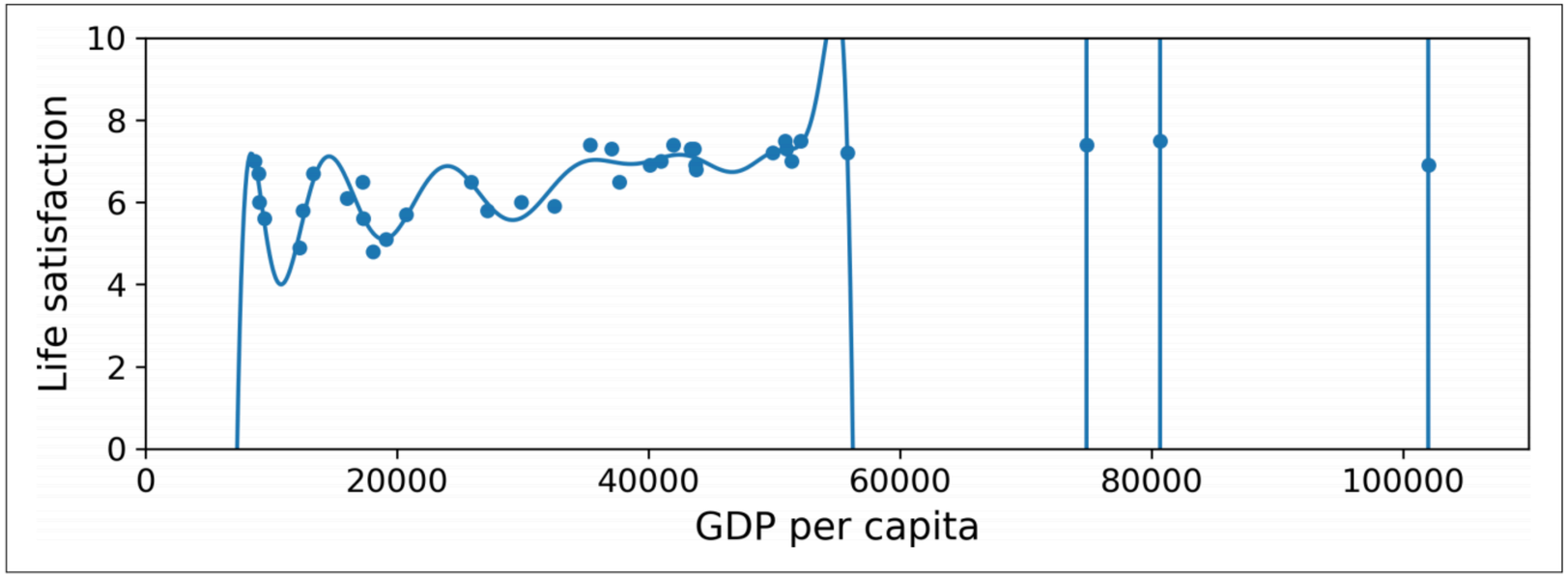
- Feature extraction: combining existing features to produce a more useful one (as we saw earlier, dimensionality reduction algorithms can help).

- Creating new features by gathering new data.

Overfitting the Training Data

- Overgeneralizing is something that we humans do all too often, and unfortunately machines can fall into the same trap if we are not careful.
- The model performs well on the training data, but it does not generalize well.

Overfitting the Training Data



Solving Overfitting

- To simplify the model by selecting one with fewer parameters.
- Constraining a model to make it simpler and reduce the risk of overfitting is called regularization.
- To gather more training data.
- To reduce the noise in the training data.

Underfitting the Training Data

- It occurs when your model is too simple to learn the underlying structure of the data.

Solving Underfitting

- Selecting a more powerful model, with more parameters
- Feeding better features to the learning algorithm
- Reducing the constraints on the model (e.g., reducing the regularization hyperparameter)