Machine Learning

1 . The Machine Learning Landscape

What Machine Learning is, what problems it tries to solve, and the main categories and fundamental concepts of its systems

HO GENT

Book pages 1-34 (printed version)

What is Machine Learning: definitions

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

More general:

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.

—Arthur Samuel, 1959

More engineering-oriented:

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

Example: SPAM filter

- = Machine Learning program
- Given:
  - Examples of spam emails (flagged by users)
  - Examples of regular (nonspam = “ham”) emails
- Learns to flag spam
- Training set = example the systems uses to learn

Why use Machine Learning?

- Traditonal approach for SPAM filter
  1. Find common words in spam emails: 4U, credit car, free, amazing, …
  2. Flag all emails that contain (a combination of) these words as spam.
  3. Test and repeat 1+2 until it’s good enough

Why use Machine Learning?

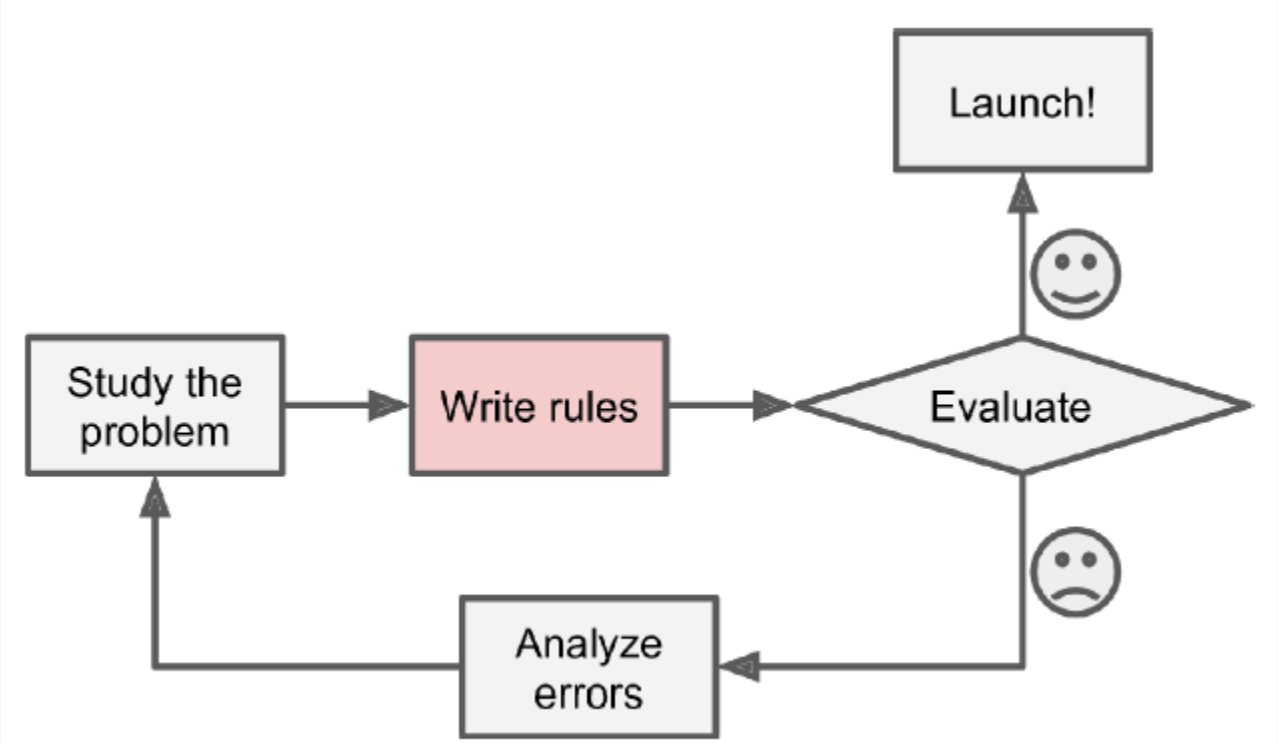


Figure 1-1. The traditional approach

HO GENT

Why use Machine Learning?

- Machine learning approach for SPAM filter
  - automatically learns which words and phrases are good predictors of spam by detecting unusually frequent patterns of words in the spam examples compared to the ham examples
  - The program is much shorter, easier to maintain, and most likely more accurate.

Why use machine learning

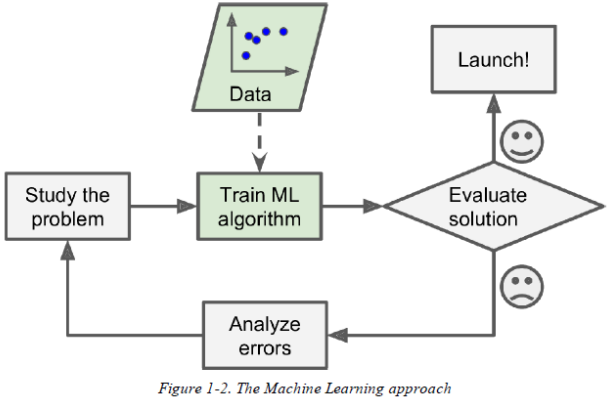


Figure 1-2. The Machine Learning approach

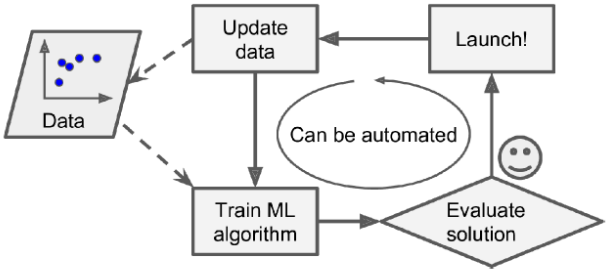


Figure 1-3. Automatically adapting to change

HO GENT

Data mining

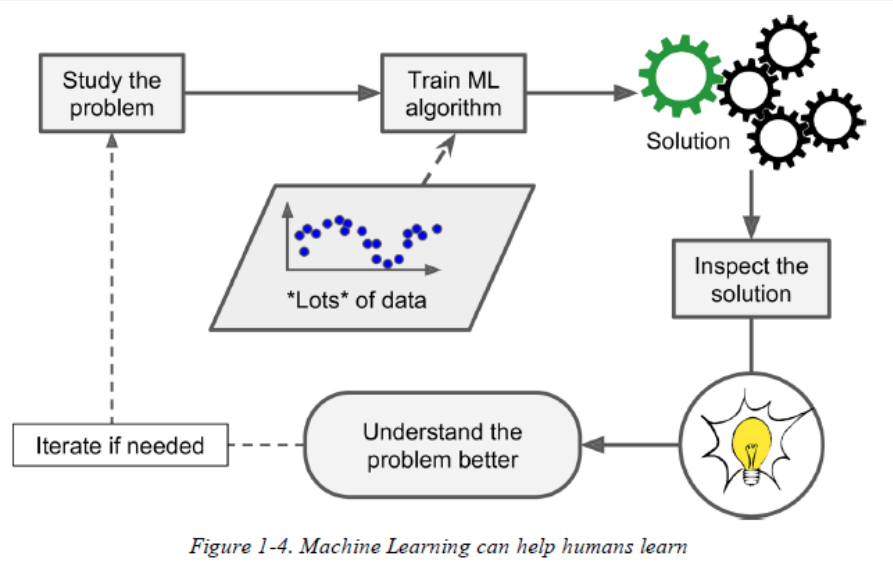


Figure 1-4. Machine Learning can help humans learn

Once a spam filter has been trained on enough spam, it can easily be inspected to reveal the list of words and combinations of words that it believes are the best predictors of spam.

Data mining = applying ML techniques to dig into large amounts of data to help discover patterns that were not immediately apparent.

HO GENT

Examples of Applications

- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans
- Automatically classifying news articles as sports, financial news, …
- 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

Types of Machine Learning systems

- Supervised/Unsupervised/Reinforcement Learning
- Batch and Online Learning
- Instance-Based vs. Model-Based Learning

Supervised/Unsupervised/Reinforcement Learning

Classified according to the amount and type of supervision they get during training.

- Supervised learning
- Unsupervised learning
- Semisupervised learning
- Reinforcement Learning

Supervised Learning: classification

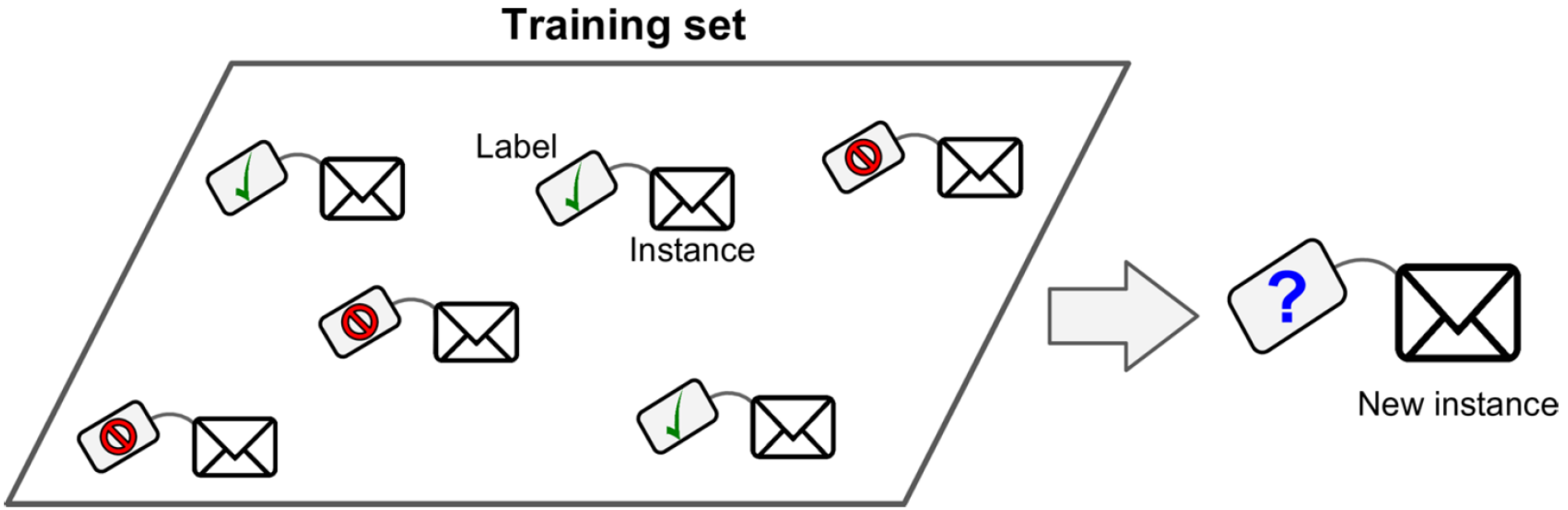


Figure 1-5. A labeled training set for spam classification (an example of supervised learning)

Supervised Learning: regression

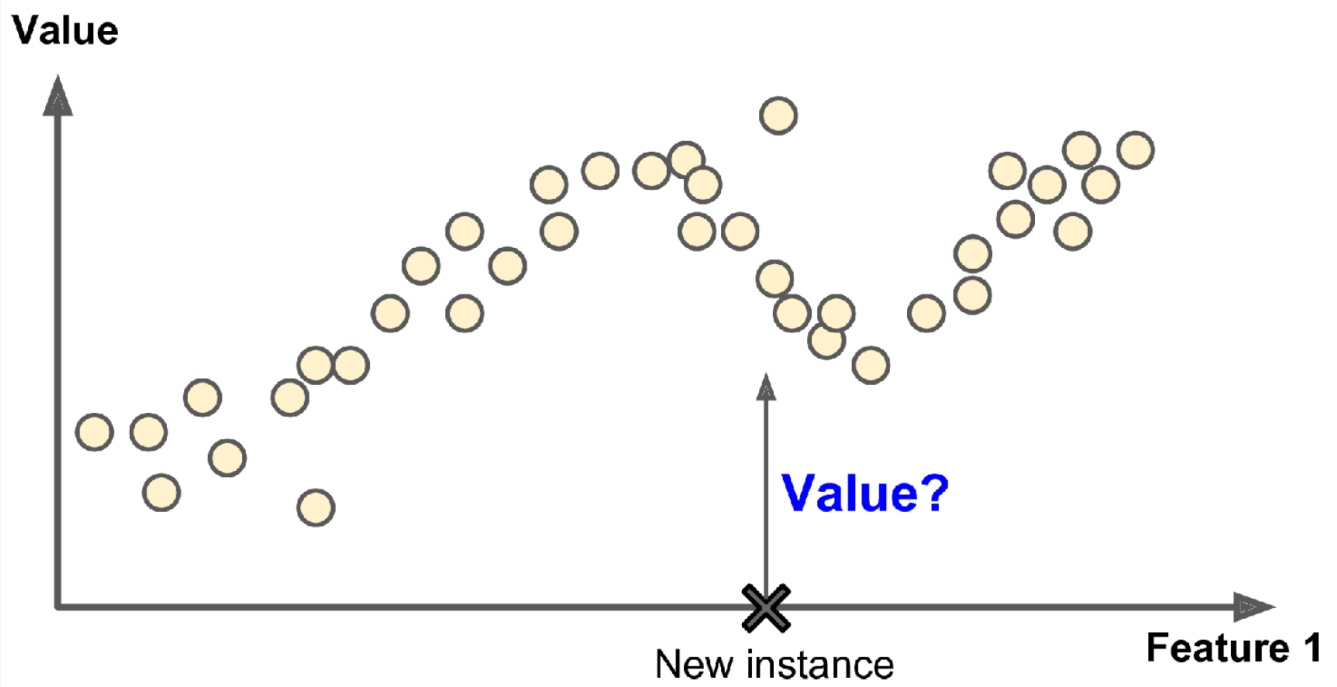


Figure 1-6. A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)

HO GENT

Supervised learning: algorithms

Some of the most important supervised learning algorithms

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Unsupervised learning

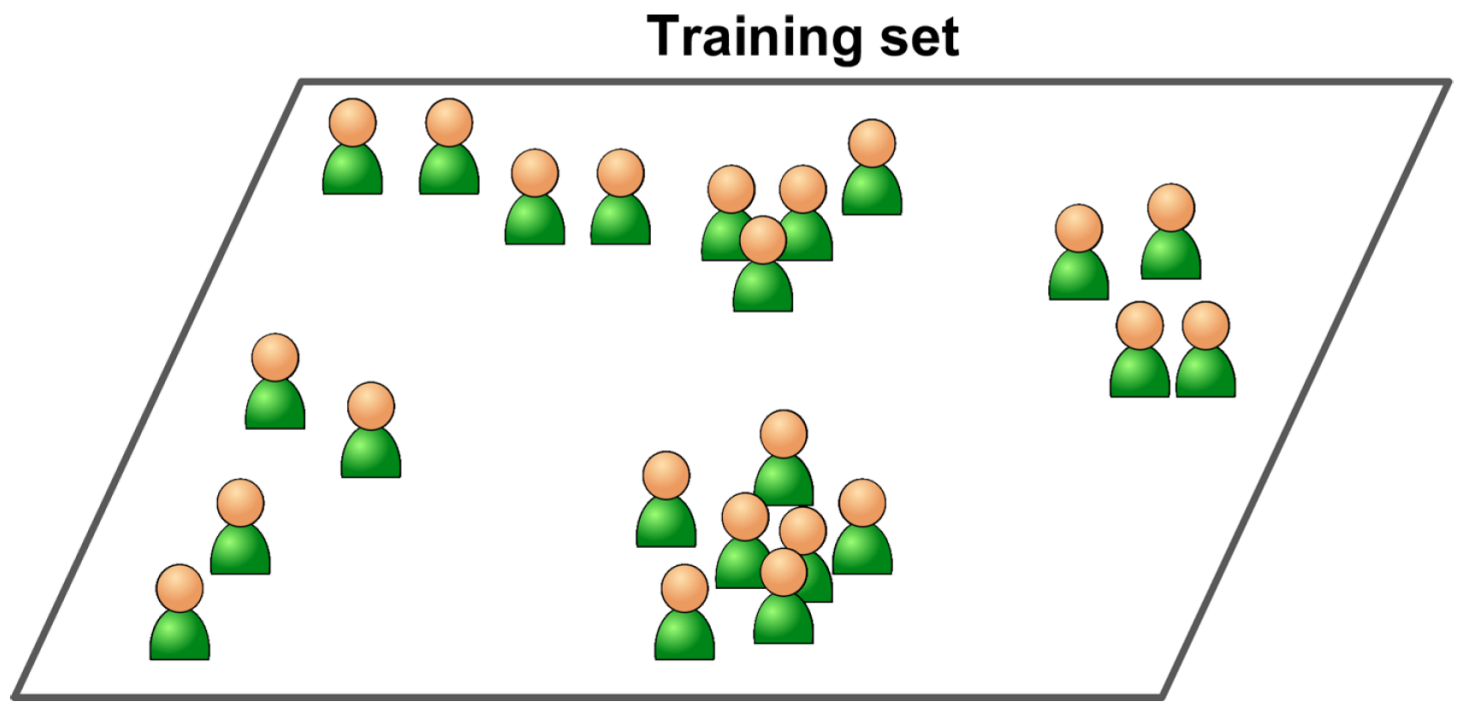
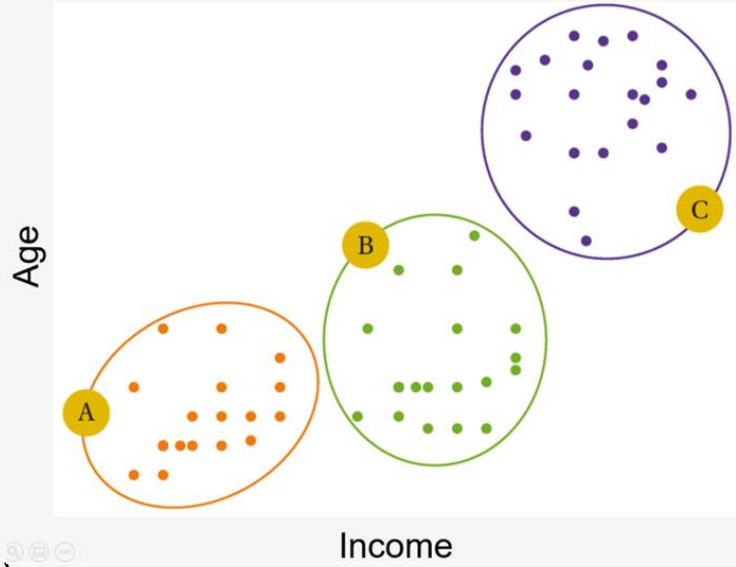


Figure 1-7. An unlabeled training set for unsupervised learning

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
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally Linear Embedding (LLE)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning (ex. Market Basket Analysis)
  - Apriori
  - Eclat



Example of clustering

HO GENT

Unsupervised learning: anomaly detection

Figure 1-10. Anomaly detection

HO GENT

Semisupervised learning

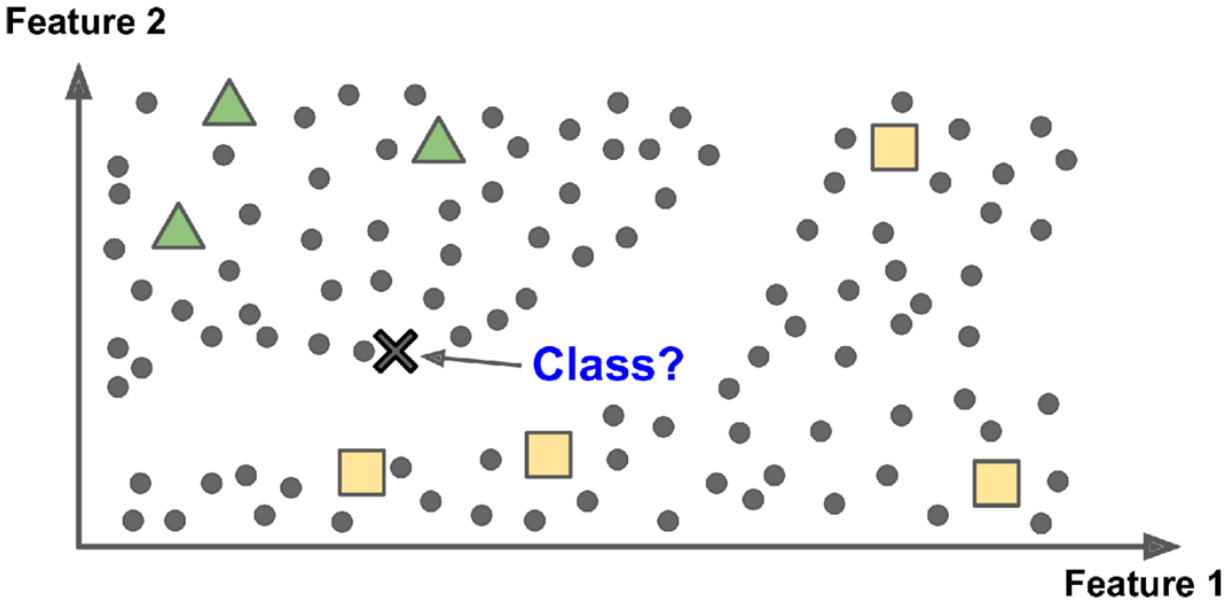


Figure 1-11. Semisupervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

HO GENT

Reinforcement Learning

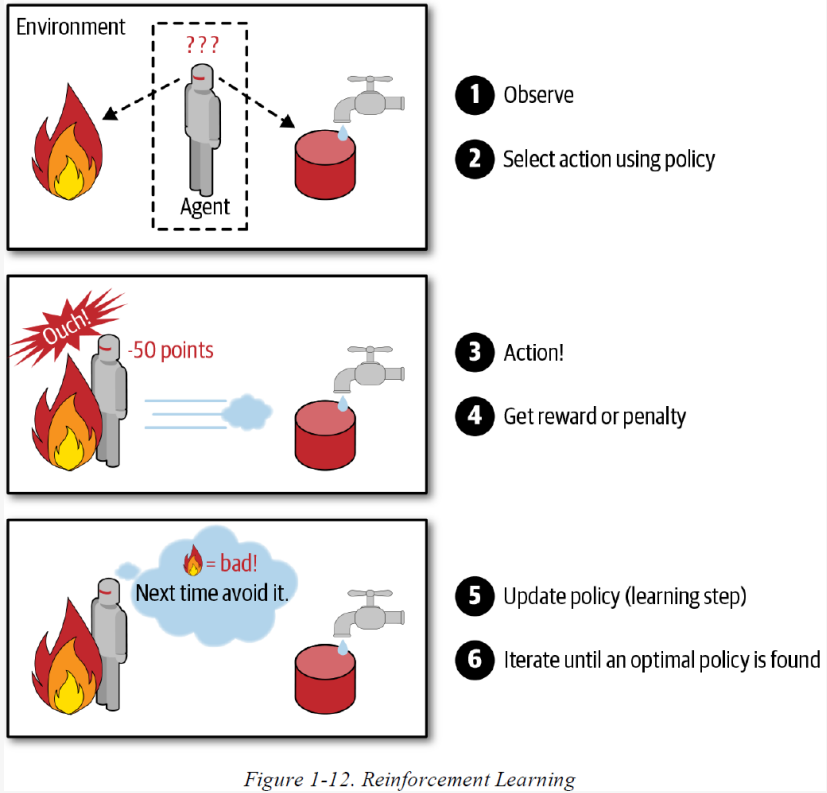
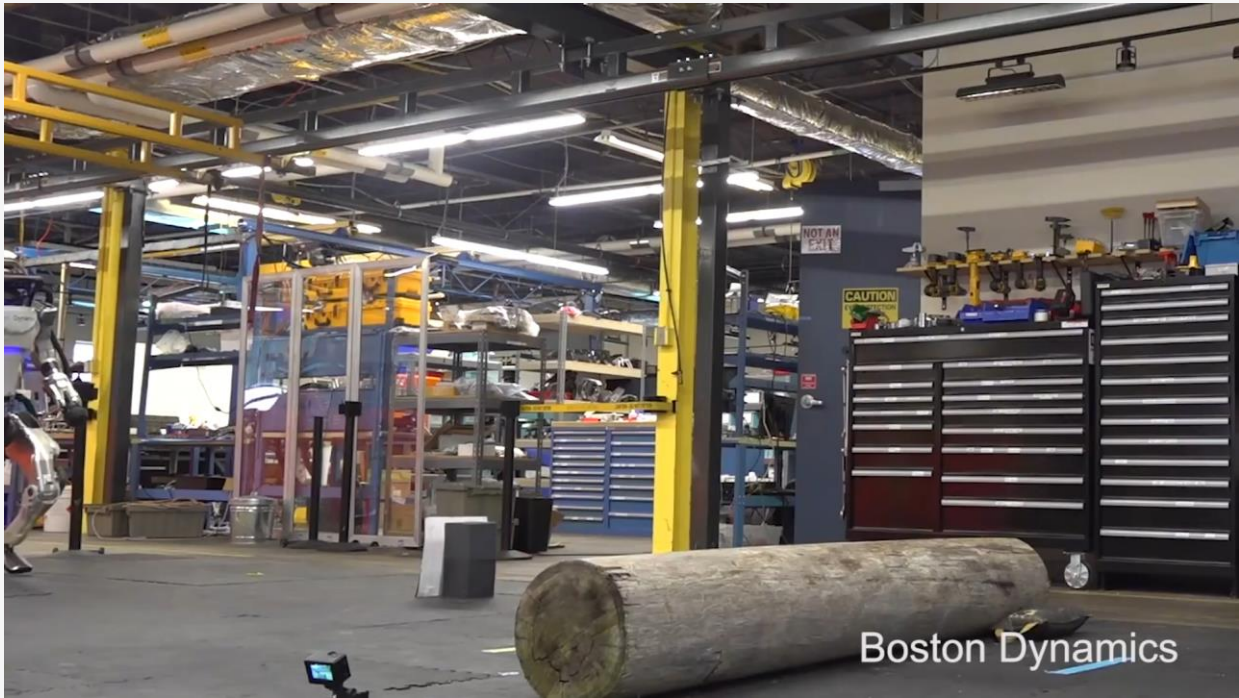


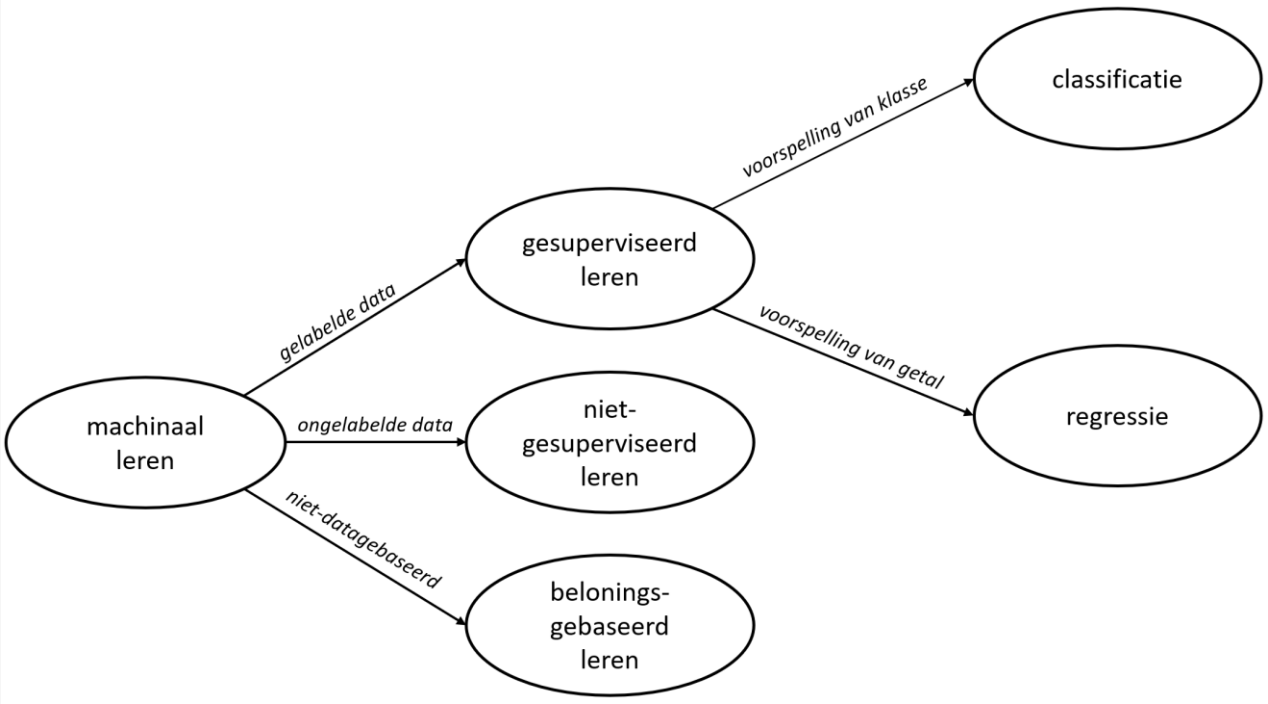
Figure 1-12. Reinforcement Learning

HO GENT

Reinforcement learning (Ndl: Beloningsgebaseerd leren): examples



Supervised/Unsupervised/ Reinforcement Learning : overview



Batch and Online Learning

- Classify Machine Learning systems based on whether or not the system can learn incrementally from a stream of incoming data

Batch learning

- System is incapable of learning incrementally
- It must be trained using all the available data
- Takes a lot of time and computing resources
  → done offline (“offline learning”)
- New data = retraining
- This process can be automated fairly easily
  (see fig. 1.3)

Online learning

- System is trained incrementally by feeding it data instances sequentially, either individually or in small groups called *minibatches*.
- Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives

Online learning

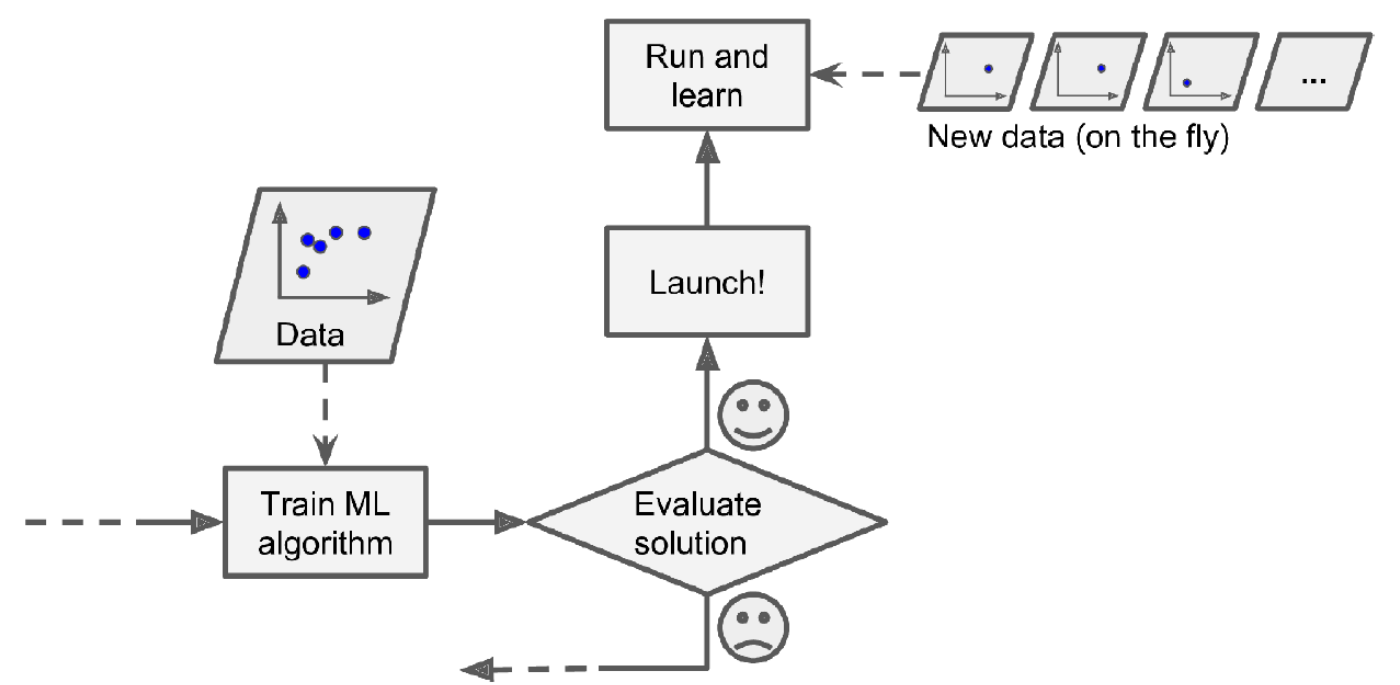


Figure 1-13. In online learning, a model is trained and launched into production, and then it keeps learning as new data comes in

HO GENT

Instance-Based vs. Model-Based Learning

- Categorize Machine Learning systems by how they *generalize*.
- Most Machine Learning tasks are about making predictions.
  - Given a number of training examples, the system needs to be able to make good predictions for examples it has never seen before. Having a good performance measure on the training data is good, but insufficient; the true goal is to perform well on new instances.
  - 2 main approaches to generalization: instance-based learning and model-based learning.

Instance based learning

- Classify according to measure of similarity
- Example: flag an email as spam if it has many words in common with a known spam email.

Instance based learning

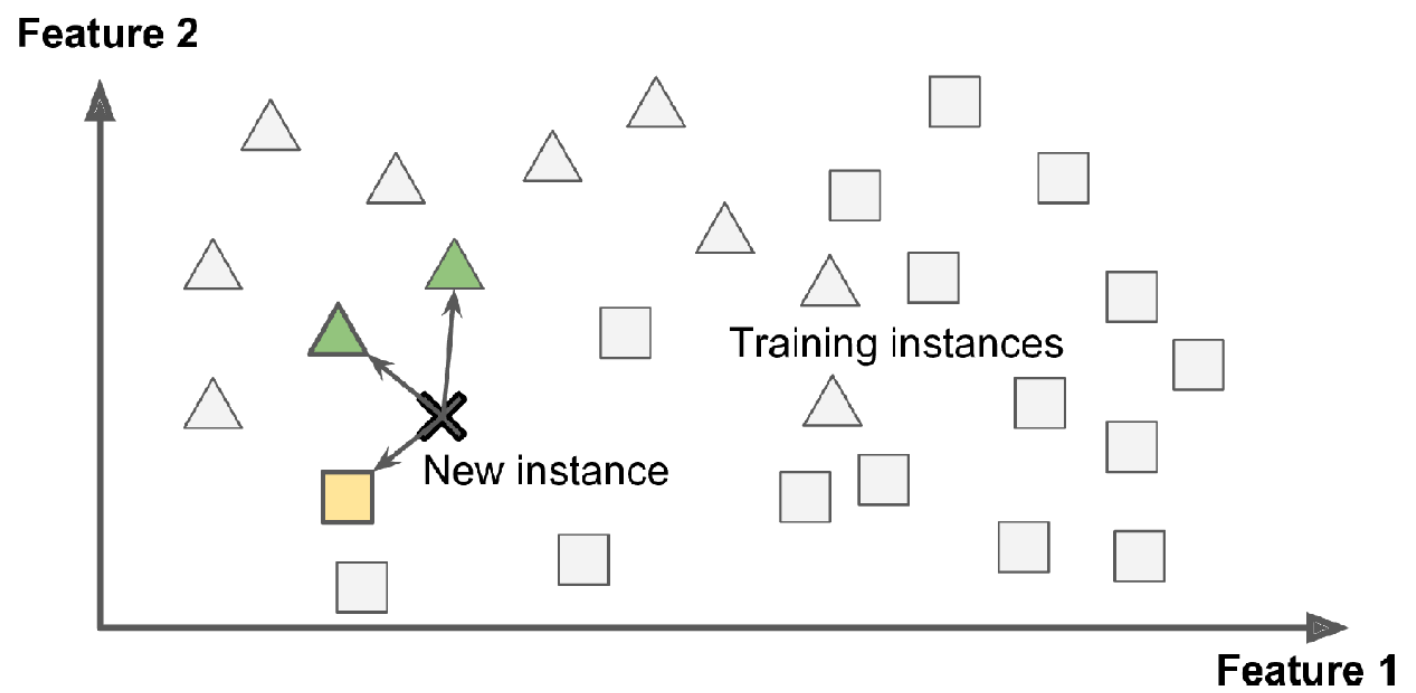


Figure 1-15. Instance-based learning

HO GENT

Model-based learning

- Build a model of these examples and then use that model to make predictions.

Model-based learning

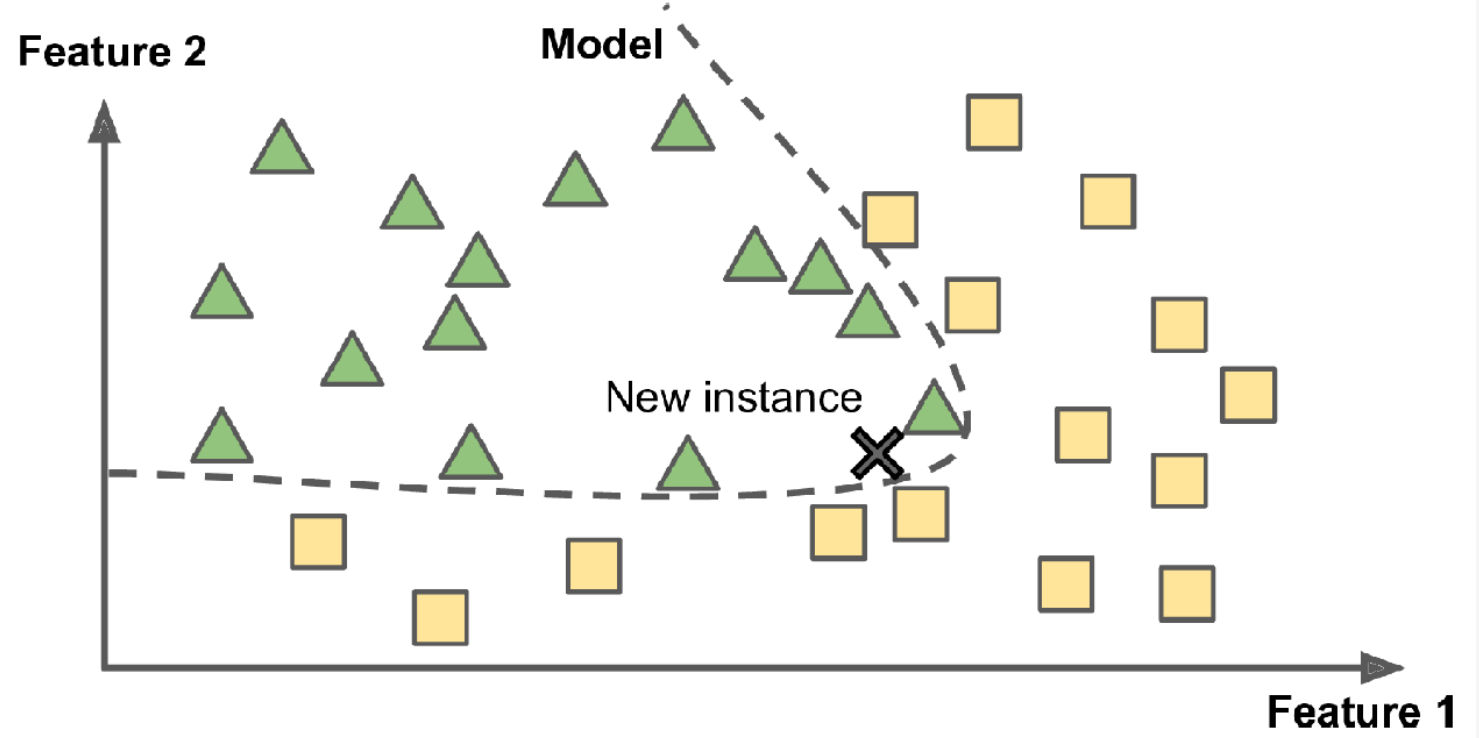


Figure 1-16. Model-based learning

HO GENT

Model Based Learning: example

- See https://github.com/jdecorte/machinelearning/blob/main/01_the_machine_learning_landscape.ipynb

Main Challenges of Machine Learning

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

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
- For complex problems such as image or speech recognition you may need millions of examples (unless you can reuse parts of an existing model).

Non-representative Training Data

- The set of countries we used earlier for training the linear model was not perfectly representative; a few countries were missing.

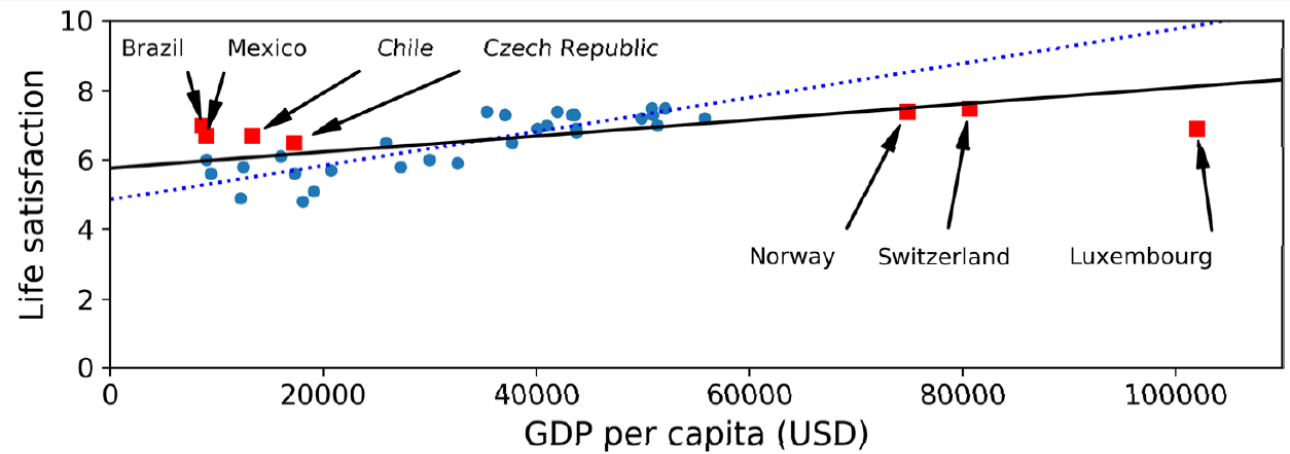


Figure 1-21. A more representative training sample

HO GENT

Non-representative Training Data

- Too few data: *sampling noise*
- Very large samples can also be nonrepresentative if the sampling method is flawed → *sampling bias*

Sampling bias: examples

- Detecting tax fraud:
  - Who is most likely to be controlled?
  - Based on previous fraud cases only detected fraud in dataset.
- Offender profiling (used e.g. in decision on provisional release)
  - Database of known offenders and their crimes
    --> offender profiles
  - What if people of certain ethnicities are over-represented in the database?

Poor-quality data

- Errors (ex. wrong classifications in training set), outliers, noise.
- First clean up the training data!
  - Some instances are clearly outliers
  - Some instances are missing a few features (e.g., 5% of your customers did not specify their age): decide whether you want to
    - ignore this attribute
    - ignore these instances
    - fill the missing values (e.g. median age)
    - train a model with the feature and one without

Irrelevant features

- Garbage in/garbage out
- 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
  - Creating new features by gathering new data

Overfitting the Training Data

- Model performs well on the training data, but it does not generalize well.

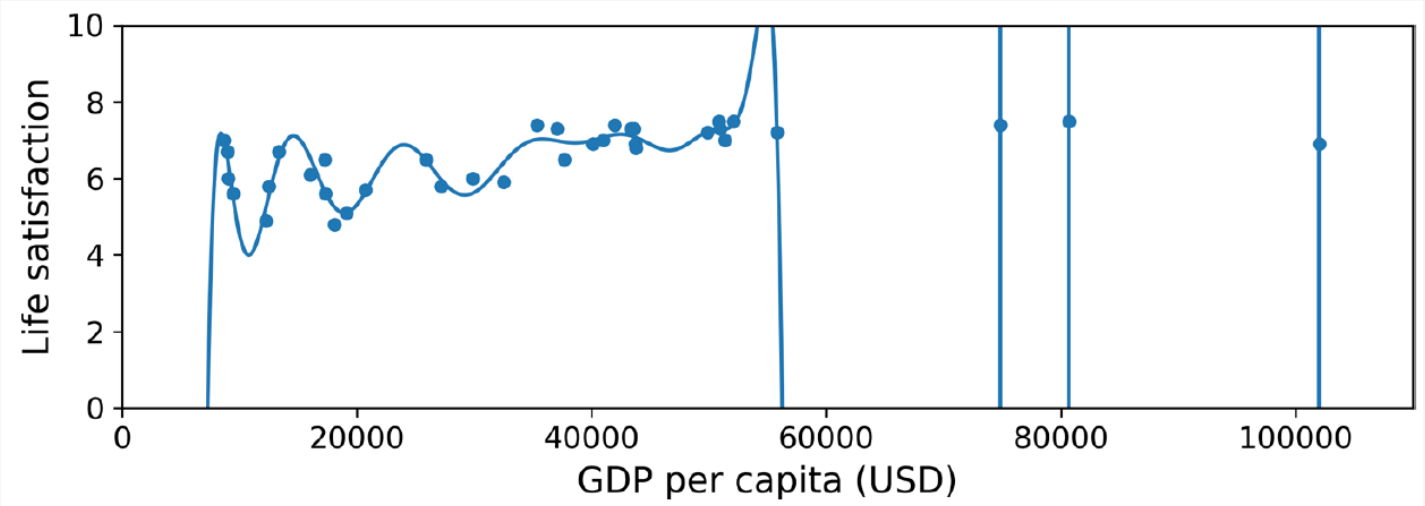


Figure 1-22. Overfitting the training data

HO GENT

Overfitting the Training Data

- Overfitting happens when the model is too complex relative to the amount and noisiness of the training data.
- The model is “learning by heart” the training samples.
- Possible solutions:
  - Simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data, or by constraining the model.
  - Gather more training data.
  - Reduce the noise in the training data (e.g., fix data errors and remove outliers).

Avoid overfitting the Training Data by regularization

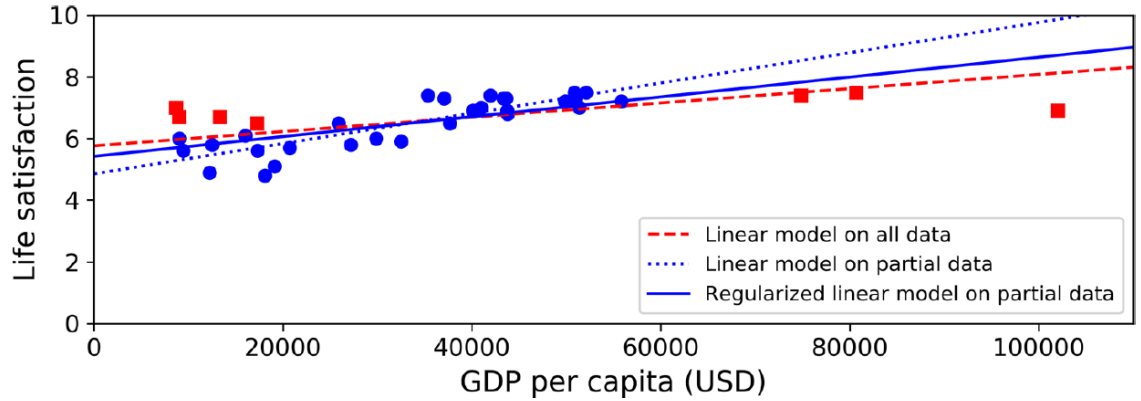


Figure 1-23. Regularization reduces the risk of overfitting

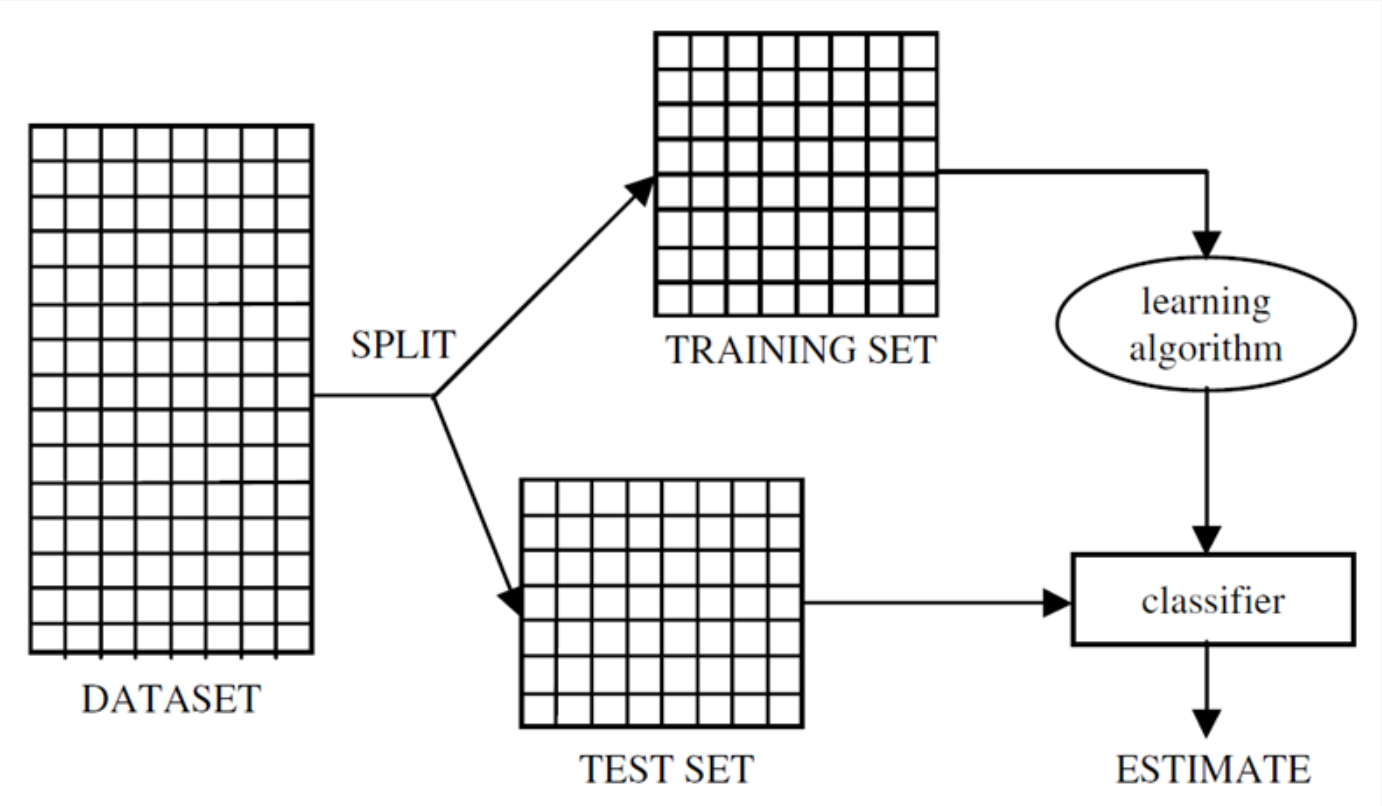
- Linear model has two parameters, θ_0 and θ_1 → 2 degrees of freedom
- $\Theta_1 = 0$: 1 degree of freedom
- Regularization: Θ_1 can vary but is forced to remain small = hyperparameter of learning algorithm

HO GENT

Underfitting the Training Data

- Opposite of overfitting
- Main options for fixing this problem:
  - Select a more powerful model, with more parameters.
  - Feed better features to the learning algorithm (feature engineering).
  - Reduce the constraints on the model (e.g., reduce the regularization hyperparameter).

Testing and Validating



Tip:
It is common to use 80% of the data for training and hold out 20% for testing. However, this depends on the size of the dataset: if it contains 10 million instances, then holding out 1% means your test set will contain 100,000 instances, probably more than enough to get a good estimate of the generalization error.

HO GENT

Hyperparameter Tuning and Model Selection

- Choosing between two types of model or different combinations of hyperparameters.
- If using same test set each time
  → optimizing for that particular test set.
  → “real” error might not be correct
- Solution:
  - use 3rd set: *holdout validation* (part of training set)
  - Train multiple models with various hyperparameters on reduced training set
  - Select best model according to validation set
  - Train this model on complete training set
  - Evaluate this model on test set

Cross-validation

- If validation set is small → imprecise evaluations
- If validation is too large → remaining training set too small → not ideal to compare candidate models
- Solution: cross validation
  - Many small validation sets
  - Each model is evaluated once per validation set after it is trained on the rest of the data
  - Average out all validations of a model
  - Drawback: training time is multiplied