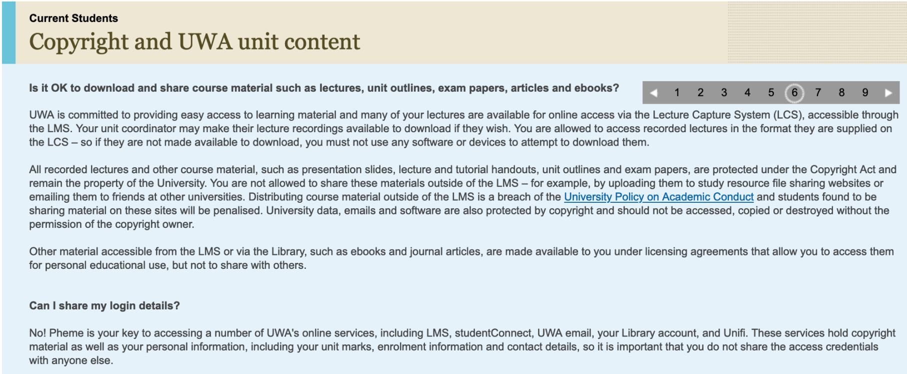


Current Students

Copyright and UWA unit content

Is it OK to download and share course material such as lectures, unit outlines, exam papers, articles and ebooks?

UWA is committed to providing easy access to learning material and many of your lectures are available for online access via the Lecture Capture System (LCS), accessible through the LMS. Your unit coordinator may make their lecture recordings available to download if they wish. You are allowed to access recorded lectures in the format they are supplied on the LCS – so if they are not made available to download, you must not use any software or devices to attempt to download them.

All recorded lectures and other course material, such as presentation slides, lecture and tutorial handouts, unit outlines and exam papers, are protected under the Copyright Act and remain the property of the University. You are not allowed to share these materials outside of the LMS – for example, by uploading them to study resource file sharing websites or emailing them to friends at other universities. Distributing course material outside of the LMS is a breach of the University Policy on Academic Conduct and students found to be sharing material on these sites will be penalised. University data, emails and software are also protected by copyright and should not be accessed, copied or destroyed without the permission of the copyright owner.

Other material accessible from the LMS or via the Library, such as ebooks and journal articles, are made available to you under licensing agreements that allow you to access them for personal educational use, but not to share with others.

Can I share my login details?

No! Pheme is your key to accessing a number of UWA's online services, including LMS, studentConnect, UWA email, your Library account, and Unifi. These services hold copyright material as well as your personal information, including your unit marks, enrolment information and contact details, so it is important that you do not share the access credentials with anyone else.

https://www.student.uwa.edu.au/learning/resources/ace/respect-intellectual-property/copyright-and-uwa-unit-content

CITS5508 Machine Learning
Introduction to Machine Learning

Débora Corrêa (Unit Coordinator and Lecturer)

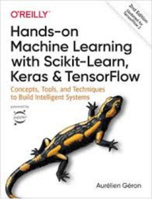
2024

Let’s get started!

Most of the slide material is from the text books: so can not be circulated and can not be used in your work (without acknowledgement).

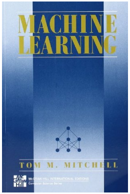
We will roughly follow this text book for the whole unit.

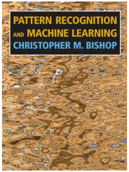
Hands-on Machine Learning with Scikit-Learn & TensorFlow

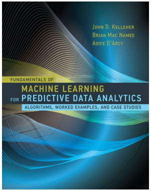

O'REILLY
2nd Edition
Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
Aurélien Géron

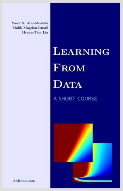
Reference books

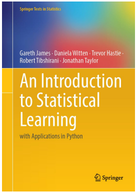
But we will also be using material from other reference books. Details in LMS.






JOHN D. KELLEHER
BRIAN MAC NAMEE
AOIFE D'ARCY
FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS
ALGORITHMS, WORKED EXAMPLES, AND CASE STUDIES




Springer Texts in Statistics
Gareth James · Daniela Witten · Trevor Hastie · Robert Tibshirani · Jonathan Taylor
An Introduction to Statistical Learning
with Applications in Python
Springer

Chapter One

The Machine Learning Landscape

- What is Machine Learning?
- Why use Machine Learning?
- Types of Machine Learning Systems
- Main Challenges of Machine Learning
- Testing and Validating

What is machine learning (ML)?

When you may be using ML on your day?

What is machine learning (ML)?

When you may be using ML on your day?

When your email services decide an email is spam, what is it doing?

What is machine learning (ML)?

When you may be using ML on your day?

When your email services decide an email is spam, what is it doing?

When Spotify suggests the next song to you, what is it doing?

What is machine learning (ML)?

When you may be using ML on your day?

When your email services decide an email is spam, what is it doing?

When Spotify suggests the next song to you, what is it doing?

What is the difference between AI and ML, or they are the same?

What is machine learning (ML)?

ML algorithms are tools for the automatic acquisition of knowledge.

ML is also known as inductive learning.

Inductive learning is a form of logical inference that allows you to obtain generic conclusions about a *particular set of examples.*

What is machine learning (ML)?

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*

Example

Let's identify these parts in the spam filter.

Experience E:

Example

Let's identify these parts in the spam filter.

Experience E:

- examples of spam emails (e.g. flagged by users)
- examples of regular/non-spam emails

Example

Let's identify these parts in the spam filter.

Experience E:

- examples of spam emails (e.g. flagged by users)
- examples of regular/non-spam emails

This is the data used to create the model.

Example

Let's identify these parts in the spam filter.

Experience E:

- examples of spam emails (e.g. flagged by users)
- examples of regular/non-spam emails

This is the data used to create the model.

Common jargon: training set, training examples, training instances, training data.

Example

Let's identify these parts in the spam filter.

Experience E:

- examples of spam emails (e.g. flagged by users)
- examples of regular/non-spam emails

This is the data used to create the model.

Common jargon: training set, training examples, training instances, training data.

The model is the learning algorithm or the inductor. Decision trees and neural networks are examples of models.

Example (cont.)

Task T:

Example (cont.)

Task T:
Flag spam for new emails.

New emails: this is the data used to verify/test the model.

Common jargon: testing set, testing examples, testing instances, testing data.

Example (cont.)

Performance P:

Example (cont.)

Performance P:

How good is my filter to identify new emails as spam or non-spam?

The performance measure P needs to be defined and will depend on the problem.

In this example, accuracy - the ratio of correctly classified emails - is one possible performance measure.

Example (cont.)

	Descriptive features				Target feature
ID	word_freq_all	word_freq_internet	word_freq_sale	⋯	Outcome
1	0.5	0.8	0.7	⋯	Spam
2	0.1	0.5	0.9	⋯	Spam
3	0.2	0.2	0.3	⋯	Regular
⋮	⋮	⋮	⋮	⋱	⋮
n	0.3	0.4	0.1	⋯	Regular

word_freq_word: percentage of words in the e-mail matching ‘word’.

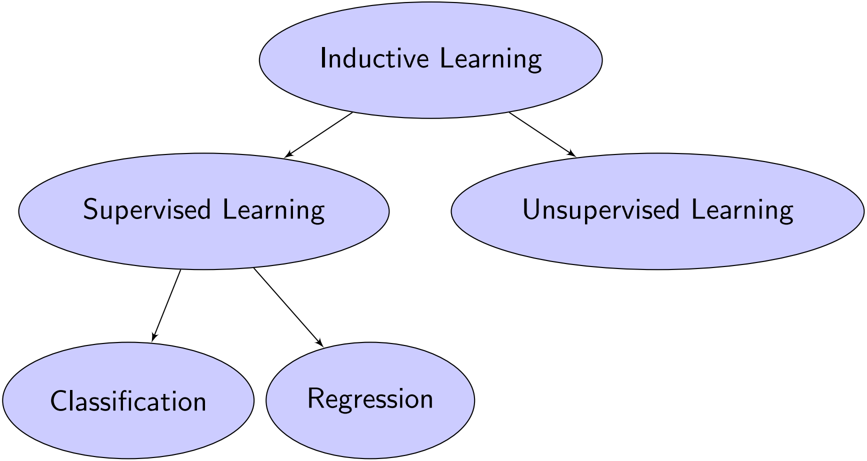
Each example/instance is described by a feature vector (or attribute vector) and by the associated label.

The whole table describes the dataset.

The objective of the inductive algorithm is to construct a classifier that can correctly assign the label of new (not labelled) examples.

Based on the data set available at `https://archive.ics.uci.edu/dataset/94/spambase`.

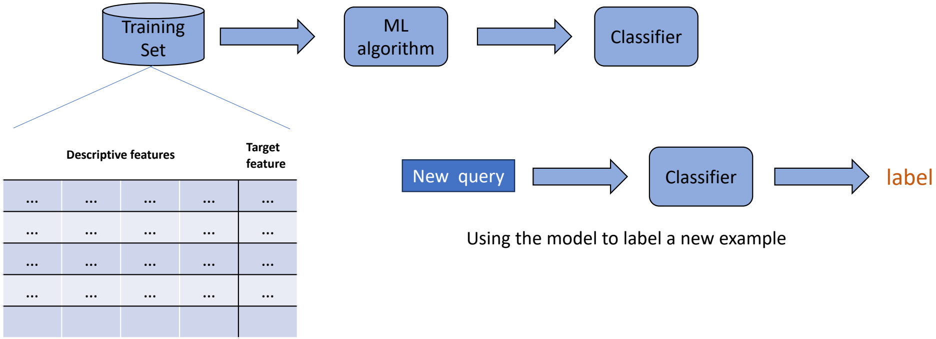
Inductive learning



We will start by looking at supervised approaches.

Two steps in supervised learning

Learning a model from a set of historical examples



Descriptive features				Target feature
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

Using the model to label a new example

Some definitions

Attribute: Also called features, predictors, independent variables. An attribute describes a characteristic or aspect of an example.

Example: Also called instance, register, data point. It is a tuple of attribute values that describes an object of interest (e.g., a regular email, a patient, or a company's customer history).

Label: Also called the target or dependent variable. It is a special attribute that describes the phenomenon of interest, i.e., the concept that the ML algorithm should learn to make predictions about it. For classification, the target variable is typically a nominal label set $\{L_1, L_2, \ldots, L_k\}$. For regression, it is a real value.

Some definitions

Data set: A data set is composed of examples with respective attribute values and the associate label. Below is a data set $\mathcal{D}$ containing n examples and m attributes.

	X_1	X_2	$\cdots$	X_m	Y
D_1	x_{11}	x_{12}	$\cdots$	x_{1m}	y_1
D_2	x_{21}	x_{22}	$\cdots$	x_{2m}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
D_n	x_{n1}	x_{n2}	$\cdots$	x_{nm}	y_n

Row D_i refers to the i-th example. That is, $i = 1, 2, \ldots, n$.

The entry x_{ij} refers to the j-th value of attribute X_j for example D_i.

Examples are tuples $D_i = (x_{i1}, x_{i2}, \ldots, x_{im}, y_i) = (\vec{x_i}, y_i)$ also denoted as (x_i, y_i), where it is implied x_i is a vector.

Some definitions - Example

Let's use another example: credit approval. The bank wishes to automate the process of evaluating credit card applications, as it receives thousands daily.

What would the examples be?

Some definitions - Example

Let's use another example: credit approval. The bank wishes to automate the process of evaluating credit card applications, as it receives thousands daily.

What would the examples be?

Customer personal information, such as annual salary, years in residence, outstanding loans, etc.

How can the bank create a target feature?

Some definitions - Example

Let's use another example: credit approval. The bank wishes to automate the process of evaluating credit card applications, as it receives thousands daily.

What would the examples be?

Customer personal information, such as annual salary, years in residence, outstanding loans, etc.

How can the bank create a target feature?

Each customer record also contains whether approving credit for that customer was a good idea.

Some definitions - Example (cont.)

The input $\vec{x}$: the customer information that is used to make a credit decision.

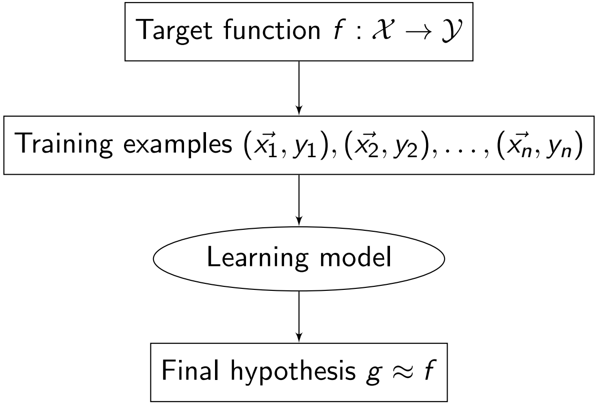
The desired unknown function $f : \mathcal{X} \to \mathcal{Y}$: ideal function for credit approval, where $\mathcal{X}$ is the input space (set of all possible inputs $\vec{x}$), and $\mathcal{Y}$ is the output space (set of all possible outputs).

The data set $\mathcal{D}$ contains all input-output examples, $(\vec{x}_i, y_i)$.

The learning algorithm uses the data set $\mathcal{D}$ to find a function $g : \mathcal{X} \to \mathcal{Y}$ that approximates f.

The function g is selected from a set of candidate functions under consideration, referred to as the hypothesis set $\mathcal{H}$. For instance, $\mathcal{H}$ could represent the set of all linear functions that the algorithm can select.

Components of learning



Target function $f : \mathcal{X} \to \mathcal{Y}$

Training examples $(\vec{x_1}, y_1), (\vec{x_2}, y_2), \ldots, (\vec{x_n}, y_n)$

Learning model

Final hypothesis $g \approx f$

The learning model is formed by the learning algorithm and hypothesis set $\mathcal{H}$.

Simple learning model

Consider $\mathcal{X} = \mathbb{R}^m$ as the input space, where $\mathbb{R}$ is the m-dimensional Euclidean space.

Let $\mathcal{Y} = \{-1, +1\}$ be the output space (binary classification).

In the credit approval example, the input vector $\vec{x} \in \mathbb{R}^m$ contains the attributes in a credit application, such as salary, years in residence, outstanding debit, etc.

The binary output y defines the decision, approving or denying credit.

The hypothesis set $\mathcal{H}$ is specified through a functional form shared by all hypothesis $h \in \mathcal{H}$.

Simple learning model (cont.)

Our functional form $h(\vec{x})$ gives different weights to the different coordinates in $\vec{x}$. This represents the relative importance of the attributes in the credit decision.

Credit is approved if $\sum_{i=1}^{m} w_i x_i > \mathit{threshold}$

Credit is not approved if $\sum_{i=1}^{m} w_i x_i < \mathit{threshold}$

Or, simply:

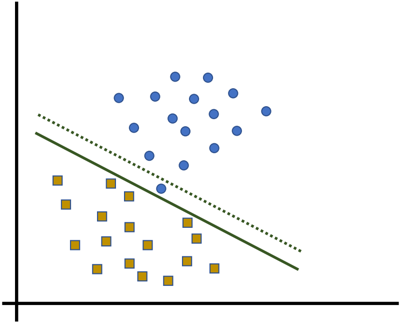
$$h(\vec{x}) = \mathit{sign}\left(\left(\sum_{i=1}^{m} w_i x_i\right) + b\right)$$

where $x_1, \ldots, x_m$ are the components of the vector $\vec{x}$, $h(\vec{x}) = +1$ means 'credit approved' and $h(\vec{x}) = -1$ means 'credit denied', $\mathit{sign}(s) = +1$ if $s > 0$ and $\mathit{sign}(s) = -1$ if $s < 0$.

Simple learning model (cont.)

Our weights are $w_1, w_2, \ldots, w_m$, and the bias term b determines the threshold as the bank will approve credit if $\sum_{i=1}^{m} w_i x_i > -b$.

Our model or hypothesis $g \in \mathcal{H}$ will be determined by the optimal choices of weights and bias.



Instance-based learning vs model-based learning

One more way to categorize Machine Learning systems is by how they generalize. Most Machine Learning tasks are about making predictions. This means that given a number of training examples, the system needs to be able to generalize to 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.

There are two main approaches ...

Instance-based learning

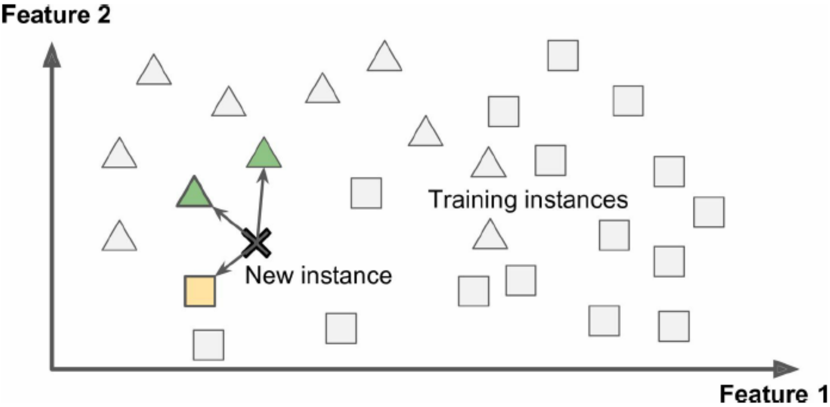


Figure 1-15. Instance-based learning

Model-based learning

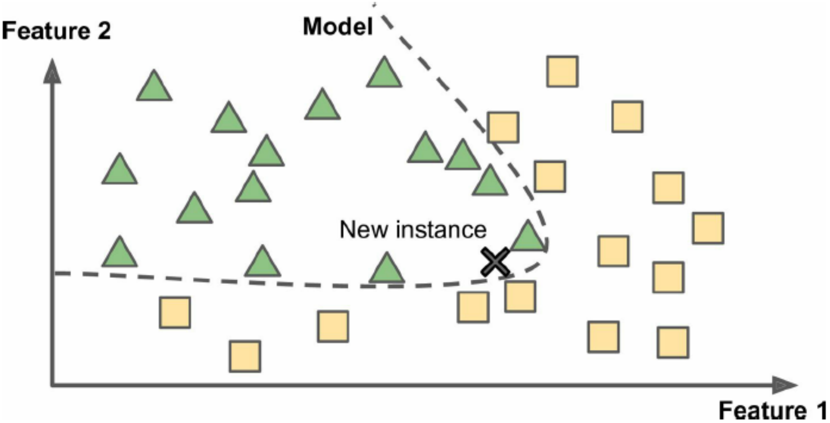


Figure 1-16. Model-based learning

Model-based learning

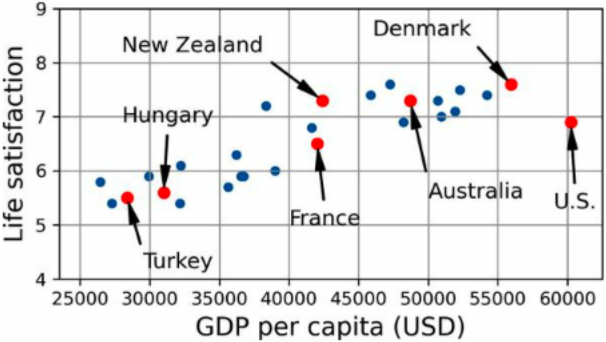


Figure 1-18. Do you see a trend here?

Model-based learning

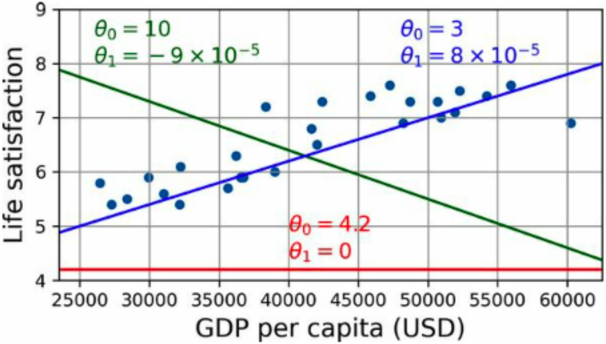


Figure 1-19. A few possible linear models

Model-based learning

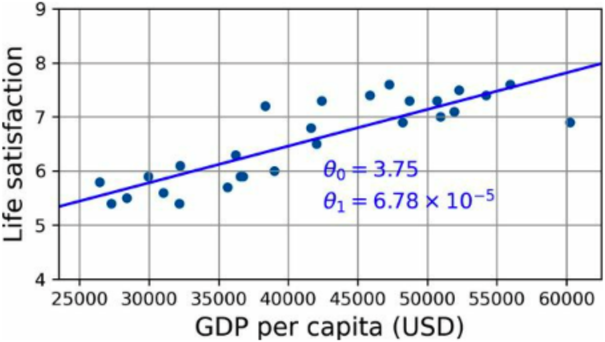


Figure 1-20. The linear model that fits the training data best

Supervised learning

Here are some of the most important supervised learning algorithms (covered in the book):

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

Unsupervised learning

The training data is *unlabelled*.

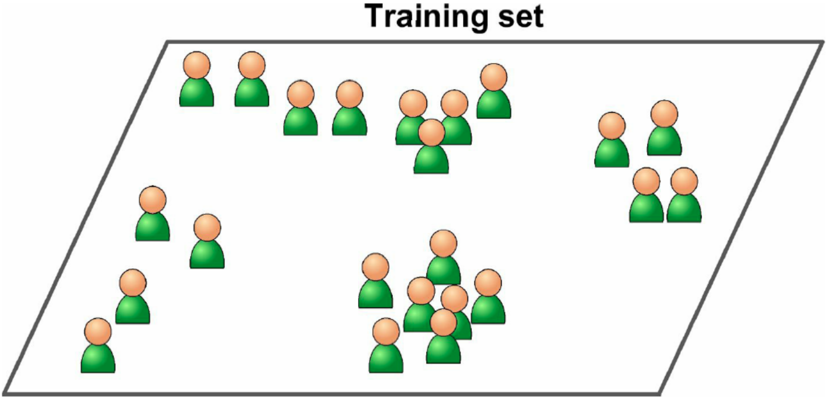


Figure 1-7. An unlabeled training set for unsupervised learning

Unsupervised learning: clustering

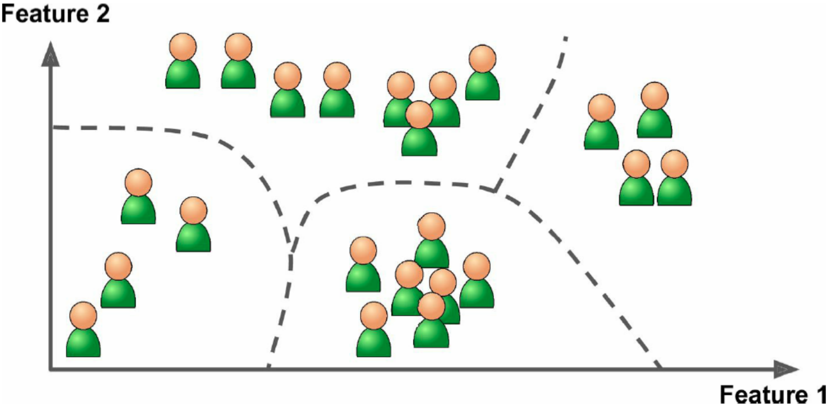


Figure 1-8. Clustering

Unsupervised learning: anomaly detection

Objective is to learn what “normal” data looks like, and then use that to detect abnormal instances.

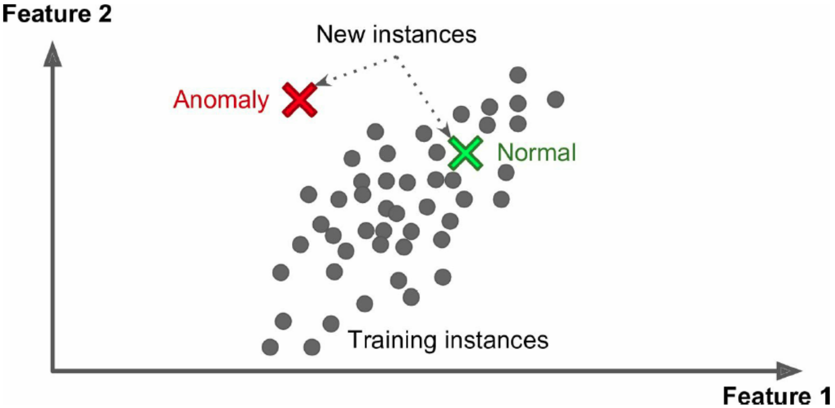


Figure 1-10. Anomaly detection

Unsupervised learning

Here are some of the most important unsupervised learning algorithms:

- Clustering
  - k-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally-Linear Embedding (LLE)
  - t-distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - Apriori
  - Eclat

Other types of machine learning systems

Many. Useful to classify them in broad categories based on:

- Whether or not they can learn incrementally on the fly (online learning versus batch learning)
- How they are supervised during training (supervised, semi-supervised, unsupervised, reinforcement learning).
- How they generalise (instance-based, model-based).

These criteria are not exclusive; you can combine them. We talked about some types today. Recommended reading includes studying about the other ones in the textbook.

Main challenges of machine learning

Main challenges of machine learning

- Insufficient quantity of training data
- Non-representative or poor-quality data
- Irrelevant features
- Overfitting or underfitting

Non-representative or poor-quality data

Some of the feature values may be incorrect, missing or may be mislabelled.

The training set is a non-representative sample of the possible instances in the domain.

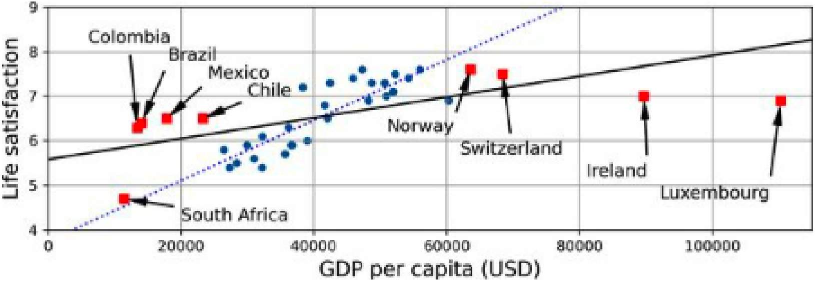


Figure 1-22. A more representative training sample

Non-representative or poor-quality data (cont.)

If your training data is poor, it will make detecting the underlying patterns harder for the system.

Much of the data scientist's effort is to understand the problem from the business perspective and prepare quality data.

You will have to deal with outliers, noise, errors, missing data, poor-quality measurements, incoherent labelling, etc.

Overfitting

The model is so complex that it fits to the dataset too closely and becomes sensitive to noise in the data. Some options:

- Simplify the model by selecting one with fewer parameters, by reducing the number of attributes, or by constraining it.
- Gather more training data.
- Check whether it is sensible to reduce the noise in the training data.

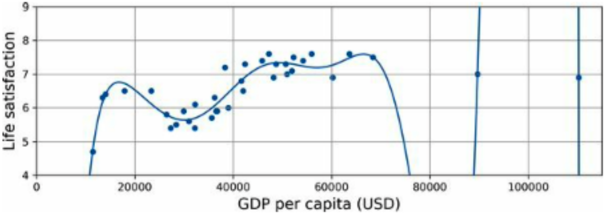


Figure 1-23. Overfitting the training data

Underfitting

Underfitting is the opposite of **overfitting**: it occurs when your model is too simple to learn the underlying structure of the data. For example, a linear model of life satisfaction is prone to underfit; reality is just more complex than the model, so its predictions are bound to be inaccurate, even on the training examples.

The main options to fix this problem are:

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

For next week

Obtain a copy of the text book and read chapter 1 and chapter 2.

Set up Python (3.X) on your computer. Write a few simple programs to be familiar with the basic syntax.

And that’s all for the first lecture.

Enjoy!