Machine Learning

Lecture 1

Justin Pearson[1]

2020

[1] http://user.it.uu.se/~justin/Teaching/MachineLearning/index.html

What is this course about?

- Understanding various machine learning algorithms including:
  - Linear Regression (as a machine learning algorithm), logistic regression, Bayesian classification, support vector machines, decision trees and clustering ...
- Common themes behind learning algorithms such optimisation by gradient descent or find parameters that maximise or minimise some measure of accuracy.
- Some practical applications.

It is important to understand what is going on behind the algorithms so that you know when to apply them.

Practical information

- Two labs in Python (Groups of 2). For the lab, I will not administer the groups via the portal. You just find a partner and hand in the required material.
- A project (Groups of 4-5). For the project I will administer the groups by the portal. You will be able to form your own groups, but if you cannot find group members then I will assign you randomly.
- An exam

We will use the scikit-learn framework. Although for some of the labs you will have to code your own algorithms.
For recommended text books please take a look at the course web page.

What is learning?

From the Oxford English Dictionary

> *Learning: The acquisition of knowledge or skills through study, experience, or being taught.*

What is Machine Learning?

> *Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel (1959)*

Machine learning grew out of Artificial Intelligence (AI), and is now considered a separate field. Machine learning has been around for a very long time.

What is Machine Learning?

> *A Well-posed Learning Problem: A computer program is said to learn from experience E with respect some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*

This is a very broad definition and could cover everything from spam filters to self driving cars to Skynet. I'll leave the philosophers in the class to work out the relationship between learning and consciousness.

Machine Learning and Data

Our focus will be statically given data.

- Our machine learning algorithm will be trained on a subset of the data (a training set)
- The performance will be then be measured by how well the algorithm predicts the correct answer on the data.

What we are not going to study

- Reactive agents. We are not going to consider an algorithm in an environment that possibly continually learn.
- No reinforcement learning.
- No biologically inspired algorithms such as genetic algorithms, ant colony optimisation or sheep herd inspired optimisation.
- We will only briefly touch on neural networks, although of our algorithms such as logistic regression are closely related to perceptrons.

Data

- Machine learning is now successful because it is easy to get hold of data.
- Large data sets can be very effective see[2]. Also AlphaGo involved training times that took weeks.
- Good data is sometimes hard to find.
- Data is not always the answer, sometimes there are algorithms out there, and as we will see later how you model a problem and what features as be as important as the data.

[2] See *The Unreasonable effectiveness of data* (Halevy, Norvig, and Pereira. `https://ieeexplore.ieee.org/abstract/document/4804817`) and *Scaling to very very large corpora for natural language* (Banko and Brill, `https://dl.acm.org/doi/10.3115/1073012.1073017`)

Statistics and Machine Learning

The relationship between statistics and machine learning is a bit complicated.

- A statistician is interested in modelling in order to understand the relationship between variables. As in machine learning the models are data driven.
- In machine learning we use data to train an algorithm in order to make predictions. Ultimately how well the algorithm does depends on how accurate predictions it makes.

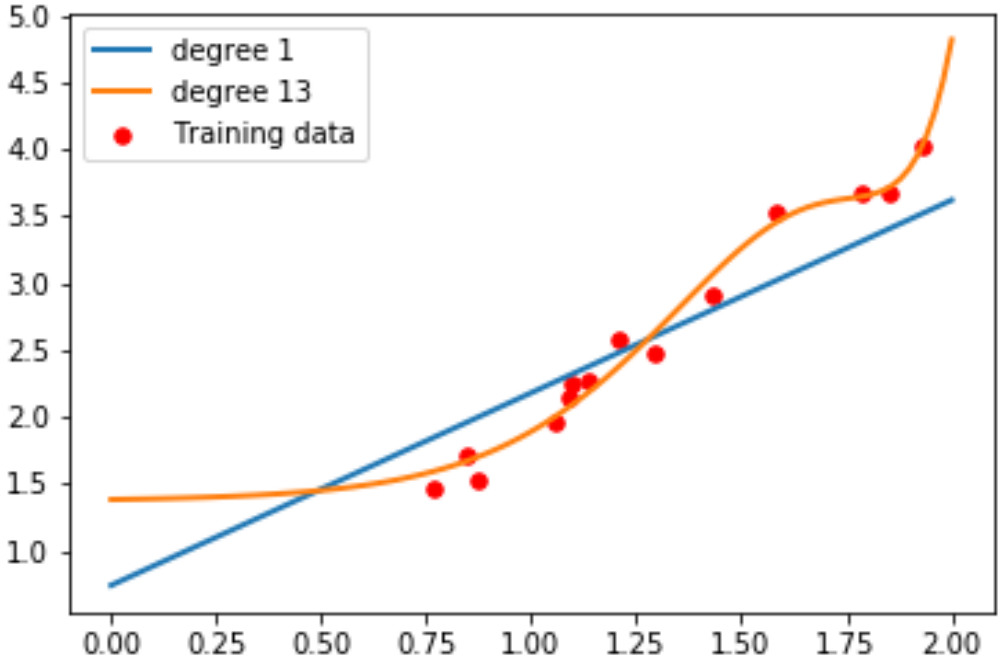
Of course there is a lot of overlap, and the two fields inform each other. A statistician will often make assumptions on the assumed distribution of data more clear than people do in machine learning.

Why might a Machine Learning perform badly?

There are lots of reasons and throughout the course we will try to understand them, but they include:

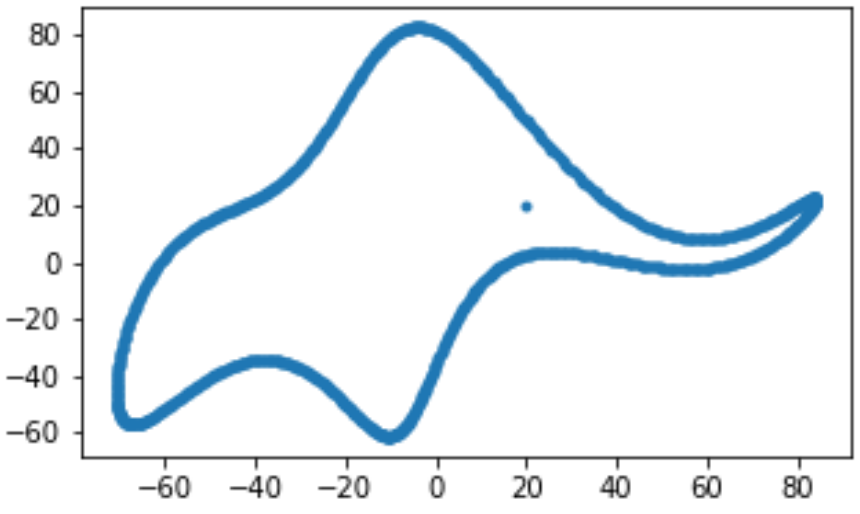
- Not enough data
- Not enough prepossessing of the data
- The wrong machine learning algorithm

Over fitting



Which model is better? Degree 1 or degree 13? If you don't have much data and you can learn complicated models you are in danger of over fitting.

Over fitting



With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

Attributed to John von Neumann.

Bias

Imagine that alien anthropologists, Zog and Zag, came to Sweden in the 1980s. After some analysis they worked out that type of car that people drove is important for their social standing. They want to analyse the different types of cars that are mod. Zog and Zag sit disguised on Strandvägen. In one hour they observe:

- 100 Volvos
- 75 Saabs

They report back to their science academy that on earth people only drive Volvos annd Saabs.
What is wrong with this picture? Sample bias, making an unwarranted generalisation from the data. They need more data to make predictions about the types of cars available on earth.

Supervised and Unsupervised Learning

Two main types of algorithms:

Supervised You are given labelled data. For each data-point you know what the correct prediction should be.

Unsupervised You just have data which is not labelled. This is given to algorithm. The most common algorithms do some sort of clustering, data-points that are similar are grouped together.

Getting good labelled data can sometimes be a problem, especially for deep neural networks where sometimes 100,000s of data-points need to gathered to train the network. Do you want to label 100,000 pictures of cats?

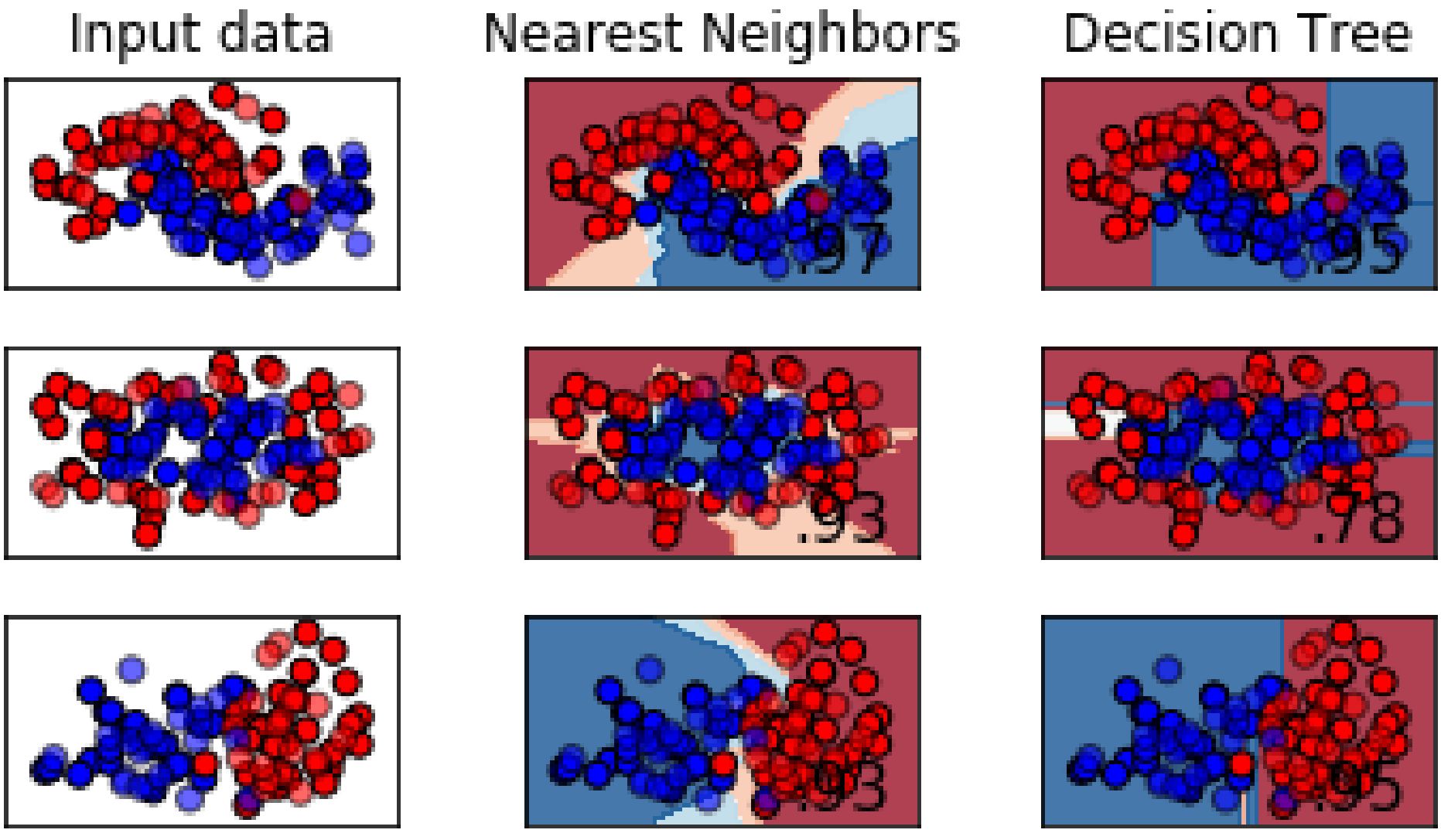
Classification and Regression

Classification Each data point should be put into one of a finite number of classes. For example email should be classified as Spam or Ham. Pictures should be classified into pictures of cats, dogs, or sleeping students.

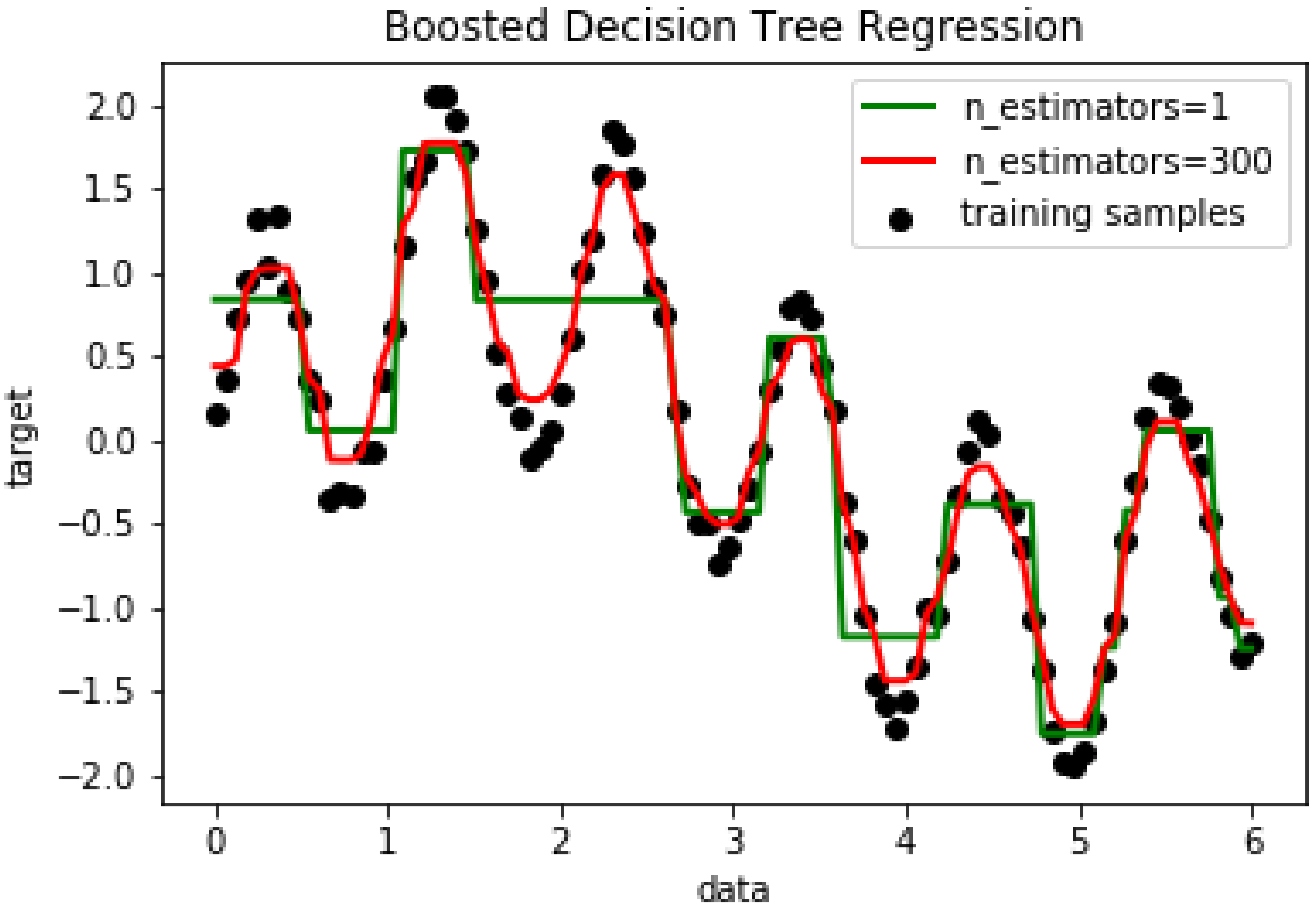
Regresssion Given the data the required prediction is some value. For example predicting house prices from the location and the size of the house.

Even though everything is finite in a computer it is easier mathematically to consider everything as a continuous variable.

Classification



Regression


Boosted Decision Tree Regression
n_estimators=1
n_estimators=300
training samples
target
data
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
1
2
3
4
5
6

How do machine learning algorithms work?

To make things easier we will concentrate on supervised learning.

- The ultimate goal of a machine learning algorithm is to make predictions.
- The algorithm learns a number of parameters given the input data. This is called the *hypothesis*.
- The goal is find a hypothesis that minimises some measure of error, sometimes called the *cost function* or the *loss function*.

Hypothesises

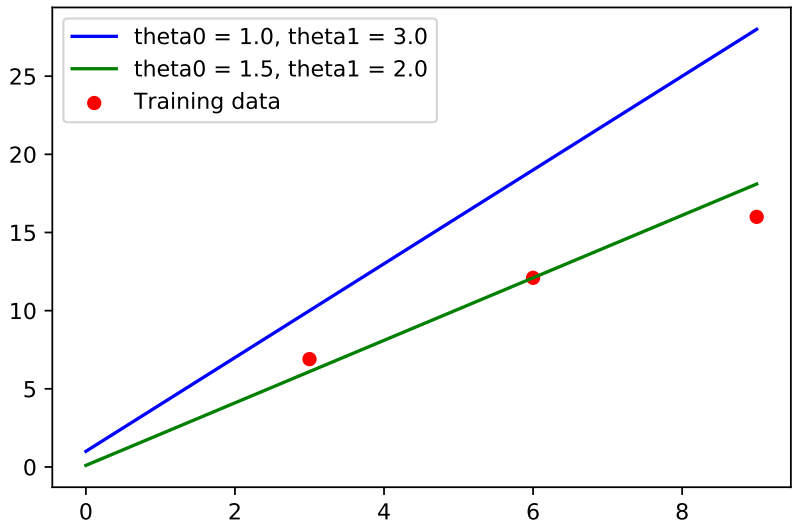
Consider a very simple data set

$$x = (3, 6, 9)$$
$$y = (6.9, 12.1, 16)$$

We want to fit a straight line to the data. Our hypothesises is a function parameterised by θ_0, θ_1

$$h_{\theta_0,\theta_1}(x) = \theta_0 + \theta_1 x$$

Hypothesises



Just looking at the training data we would say that the green line is better. The question is how to we quantify this?

Measuring Error - RMS

Root Mean Squared is a common cost function for regression. In our case given the parameters θ_0, θ_1 the RMS is defined as follows

$$J(\theta_0, \theta_1, x, y) = \frac{1}{2m} \sum_{i=1}^{m} (h_{\theta_0,\theta_1}(x^i) - y^i)^2$$

We assume that we have m data points where x^i represents the ith data point and y^i is the ith value we want to predict. Then $h_{\theta_0,\theta_1}(x^i)$ is the model's prediction given θ_0 and θ_1.
For our data set we get

$$J(1.0, 3.0) = 33.54$$
$$J(1.5, 2.0) = 2.43$$

Obviously the second is a better fit to the data.
Question why $(h_\theta(x) - y)^2$ and not $(h_\theta(x) - y)$ or even $|h_\theta(x) - y|$.

Learning

The general form of regression learning algorithm is as follows:

- Given training data $x = (x^1, \ldots, x^i, \ldots, x^m)$ and $y = (y^1, \ldots, y^i, \ldots, y^m)$
- A set of parameters Θ where each $\theta \in \Theta$ gives rise to a hypothesis function $h_\theta(x)$;
- A loss function $J(\theta, x, y)$ the computes the error or the cost for some hypothesis θ for the given data x,y;
- Find a (the) value θ that minimises J.

How we do this will be the topic of a later lecture.

Classification

Remember two classes of machine learning algorithms: Regression where we want to predict a value (or multiple values) and Classification where we want to predict which class the data belongs to.
Examples of classification problems include:

- Email spam detection, is my current email spam or ham?
- Given some medical information such as the size of a tumour is the tumour cancerous or not.
- Given an image is it a cat, dog or a horse?

Classification — Representation

Typically classes are represented by integer variables.

- $Y = 0$ or $Y = 1$ where $Y = 1$ means that it is spam.
- $Y \in \{0, 1, 2\}$ where 0 means a cat, 1 means a dog and 2 means a horse.

Using RMS to measure classification error would not make much sense (Why?)

Classification — Measuring Error

One approach is to use probability. We develop an algorithm that gives a probability that the input data is in a particular class.
We then want to maximise the probability that algorithm makes true predictions.

Classification — Confusion Matrices

Consider a classifier that tries to predict if an image is Cat or a non-Cat (everything else).
Given some input x, four things can happen:

True Positive x can be a cat and we predict a cat

False Negative x is a cat, but we predict a non-cat

False Positive x is not a cat, but we predict a cat

True Negative x is not a cat, and we predict non-cat.

True Positive and True Negative are the good things. We want to minimise False Positives and False Negatives. Sometimes we cannot minimise both.

Classification — Confusion Matrices

We can put this into a table:

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Classification — Confusion Matrices

What do we minimise? False positive or false negative?

- In the tumour example, a false positive means we predict a tumour but there was none. If this is the case we just have to investigate more to find the true situation. We want to avoid false negatives where there is a tumour, but the algorithm predicts that there is none.
- If our spam detector classifies as spam and non-spam. A false positive means that a non-spam message gets classified as spam. This means that you might miss important email that is put into the spam filter.

Later on we will look at how to tune classification algorithms to give different confusion matrices.

Training and Test Sets

One of the most important things to remember in machine learning. You must divide that data into two parts:

Training Set This is the data you use to find the best parameters of the model or hypothesis. Machine learning can be seen as an optimisation problem find the parameters that best explain the data under some error/cost or loss function.

Test Sets The test set is what you use to validate your model. You are interested in the error/cost on this set.

In scikit-learn there is a function `train_test_split` that does randomised splitting of the data. You can decide what percentage of the data is used for training.

Training and Test Sets

- Why is a bad idea to train on the whole data set?
  - One of the main reasons is that you want to avoid over fitting.
- What is the main factor that might affect what percentage of the data is used for training?
  - How much data you have. If you have too little data then you algorithm will biased towards your data set and your conclusions will not generalise.

  Later on we will look at cross validation which splits up the data into multiple training and test sets.

To come

- Revision of basic probability
- Naive Bayesian Classifiers
- Some very basic calculus revision: gradients, minimums and gradient descent.
- Some basic regression algorithms.

Not a lecture slide — Concepts covered

Learning, definition of Machine Learning, data, over fitting, bias, supervised and unsupervised learning, classification, regression, error/cost or loss functions, hypothesises, find the best hypothesises that explains the data by minimising the cost function, confusion matrices, training and test sets.