Machine Learning Basics

Agenda

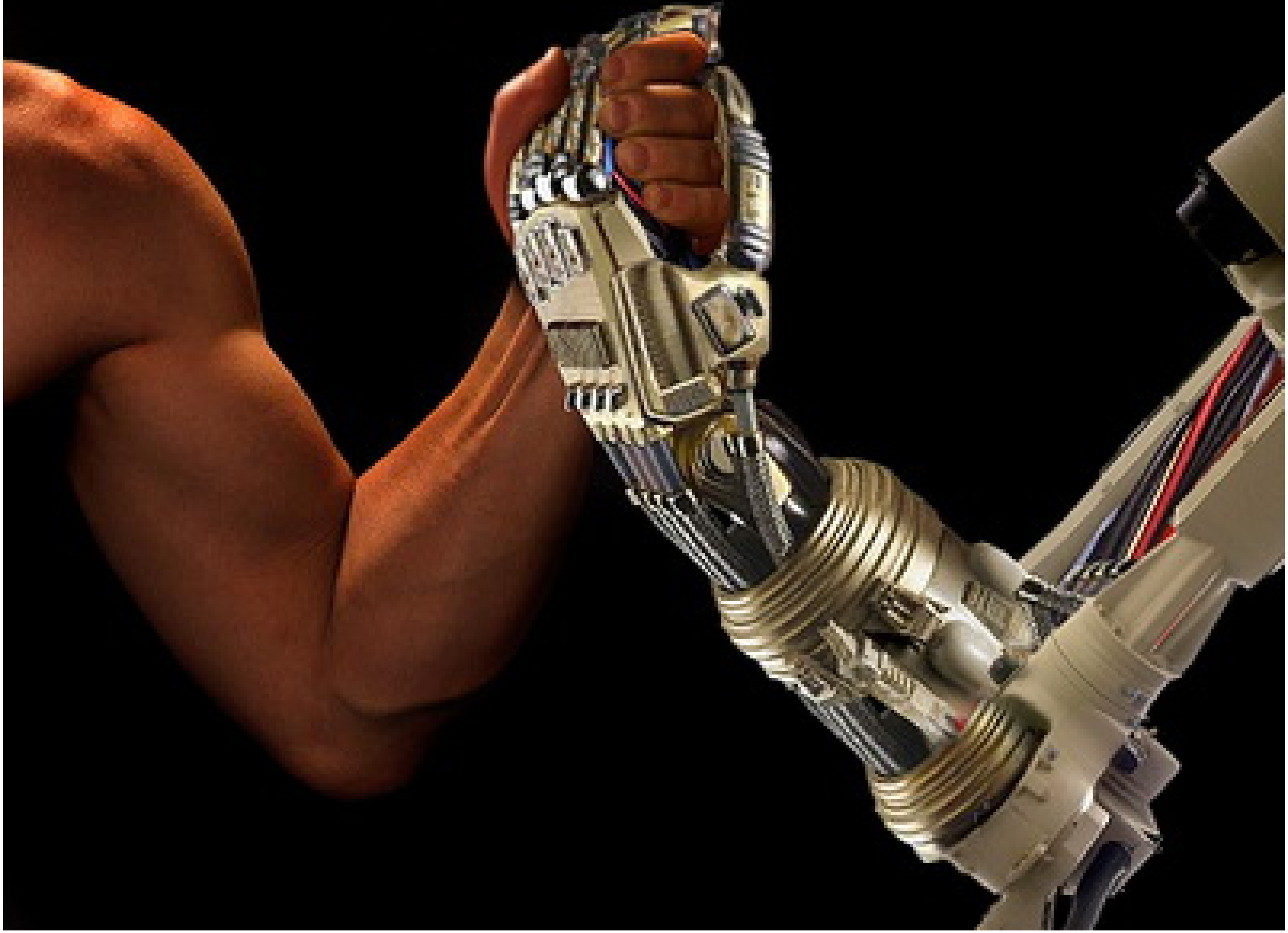
- What is Machine Learning ?
- Classification
- Classification Algorithms
    - k-nearest Neighbors
    - Support Vector Machines
    - Decision Tree
- Regression
- What's next ?

Machine Learning Definition

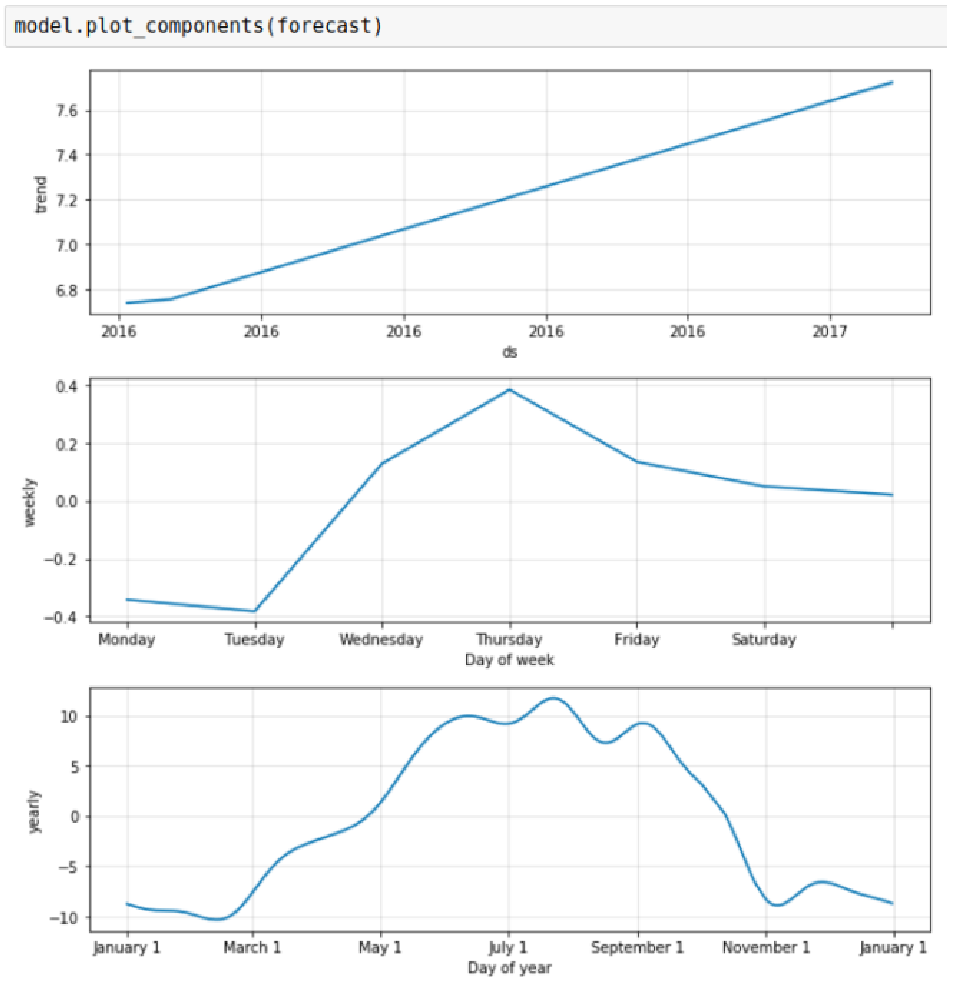
"A computer program is said to learn to perform a task T from experience E, if its performance at task T, as measured by a performance metric P, improves with experience E over time."

"Machine Learning", Tom Mitchell, 1997

Statistics vs Machine Learning

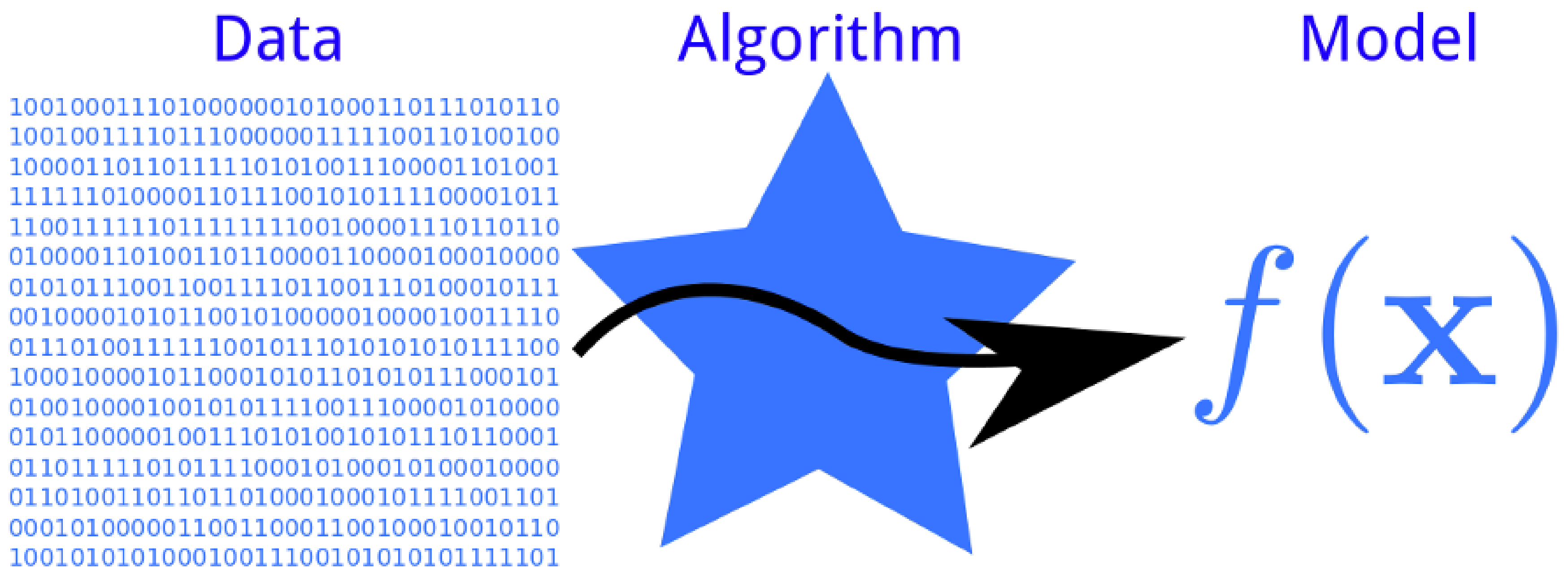


Statistics



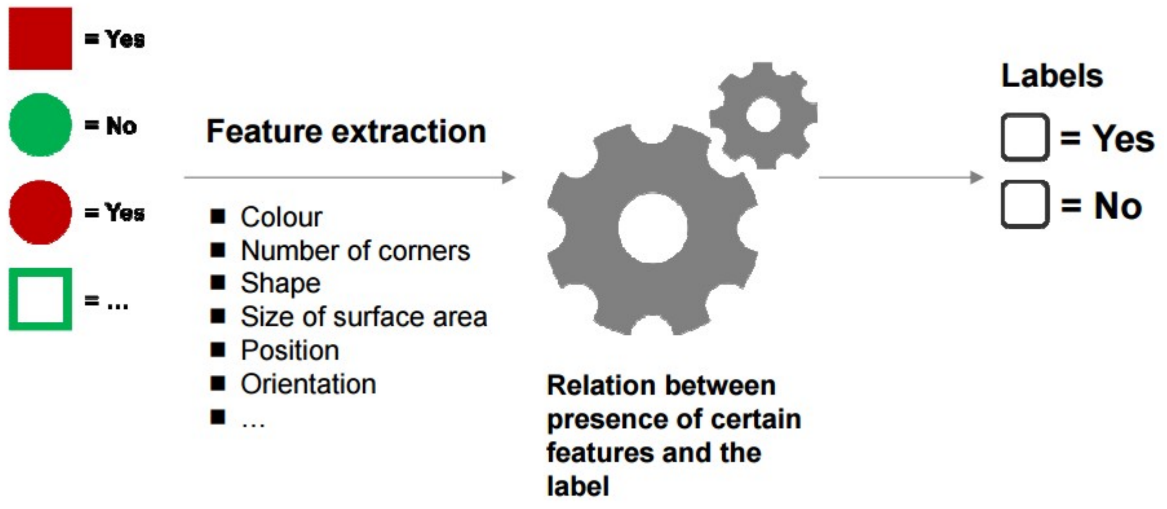
```
model.plot_components(forecast)
```

Machine Learning



How do machines learn ?

1 The first type of learning

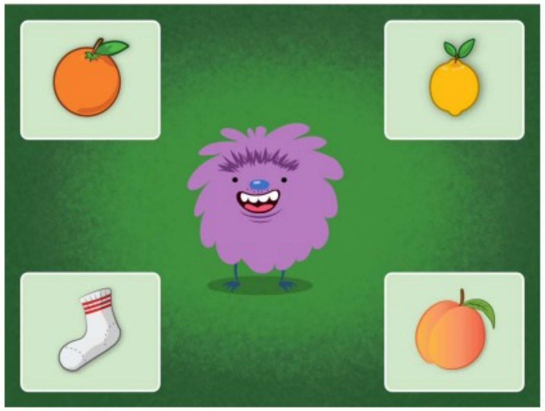


1 By generalising
from examples + correct answers

- You are given examples with the correct answer
- From this you infer some form of rules (you generalise)
  - Actually, before you infer the rules, you extract *features* (like colour, shape, number of edges, etc.)
- Then you apply these rules to a new example in order to predict the answer
- If you get a new correct answer, you can correct your rules and get even better
- Remarks:
  - Do I have enough examples to derive the relation?
  - Have I considered the right 'features' to derive the relation?

2 The second type of learning

Please create groups of similar keys.



Source: education.com

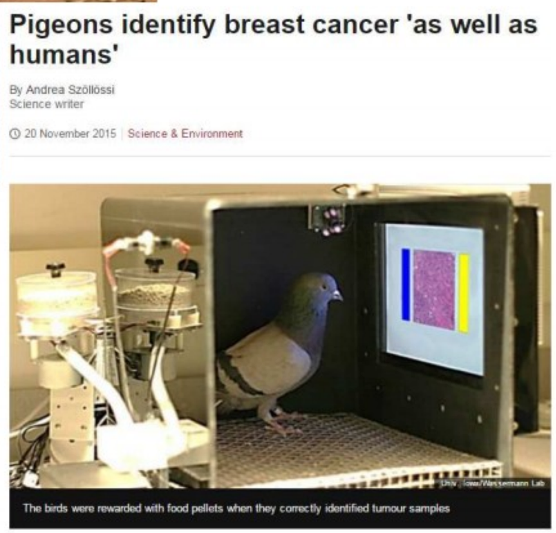
One of the things is not like the others.
Find the thing that doesn't belong.

2 By comparing

- You look at the things around you, compare them, arrange them according to similarity and then gain some insights (groups of similar items, odd ones, somehow important ones …)
- For this, you do not need the „right" answer; it might even be difficult to define the "right" answer
- Remarks:
  - Do I have enough examples to understand what similar means?
  - Do I consider the right things (the right *features*) when I say two things are similar?
  - How do I know how many groups you want?

3 The third type of learning

You can find the video about rats sniffing out land mines here: https://www.youtube.com/watch?v=nEm5zR1IND0



Pigeons identify breast cancer 'as well as humans'

By Andrea Szöllössi
Science writer

20 November 2015 | Science & Environment

The birds were rewarded with food pellets when they correctly identified tumour samples

You can find the video about pigeons identifying cancer here: https://www.youtube.com/watch?v=flzGjnJLyS0

3 By feedback (reward signal)

- I don't tell you what or how to do it.
- I don't give you any examples at the beginning.
- But I will tell you after you have done something good (delayed feedback)
- I might also tell you *how* good you have been (smaller or bigger award)
- So I use some reward to reinforce behaviour that should be maintained or increased
- Other examples: Learning how to walk, riding a bike, …
- Remarks:
  - Ok, this takes ages.
  - How complex can the behaviour get if you just get a reward signal? → Very complex.

Which of the following problems is it ?

Face recognition
("Who is the person on this photo?")

Customer segmentation
("What types of customers do we have?")

House price estimation
("How much is this house worth?")

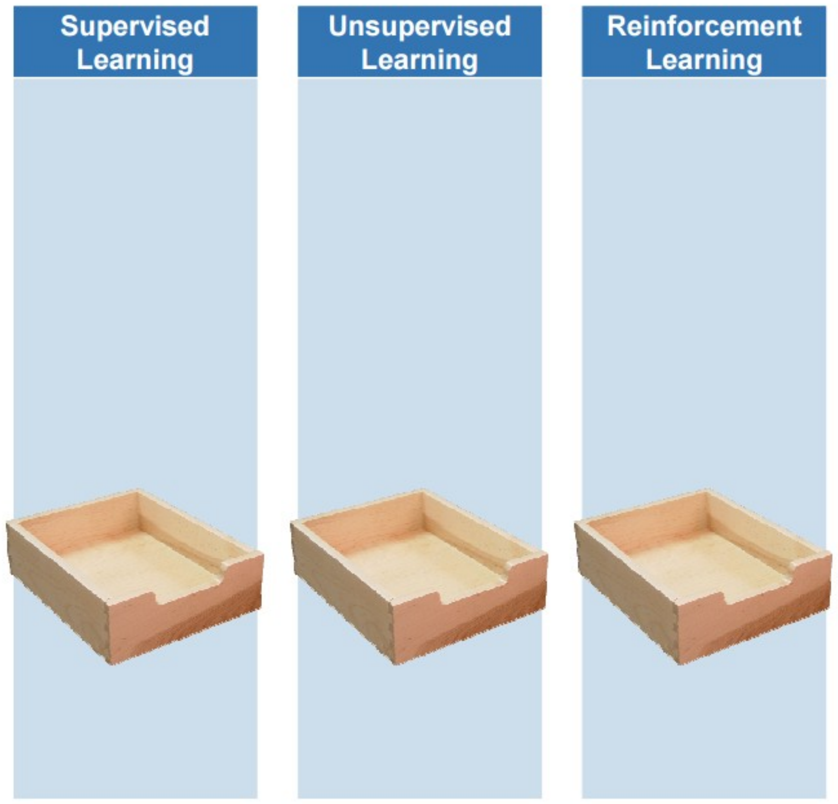
Fraud detection
("Is there anything fishy going on with this client's credit card transactions?")

Spam filter
("Is this email spam?")

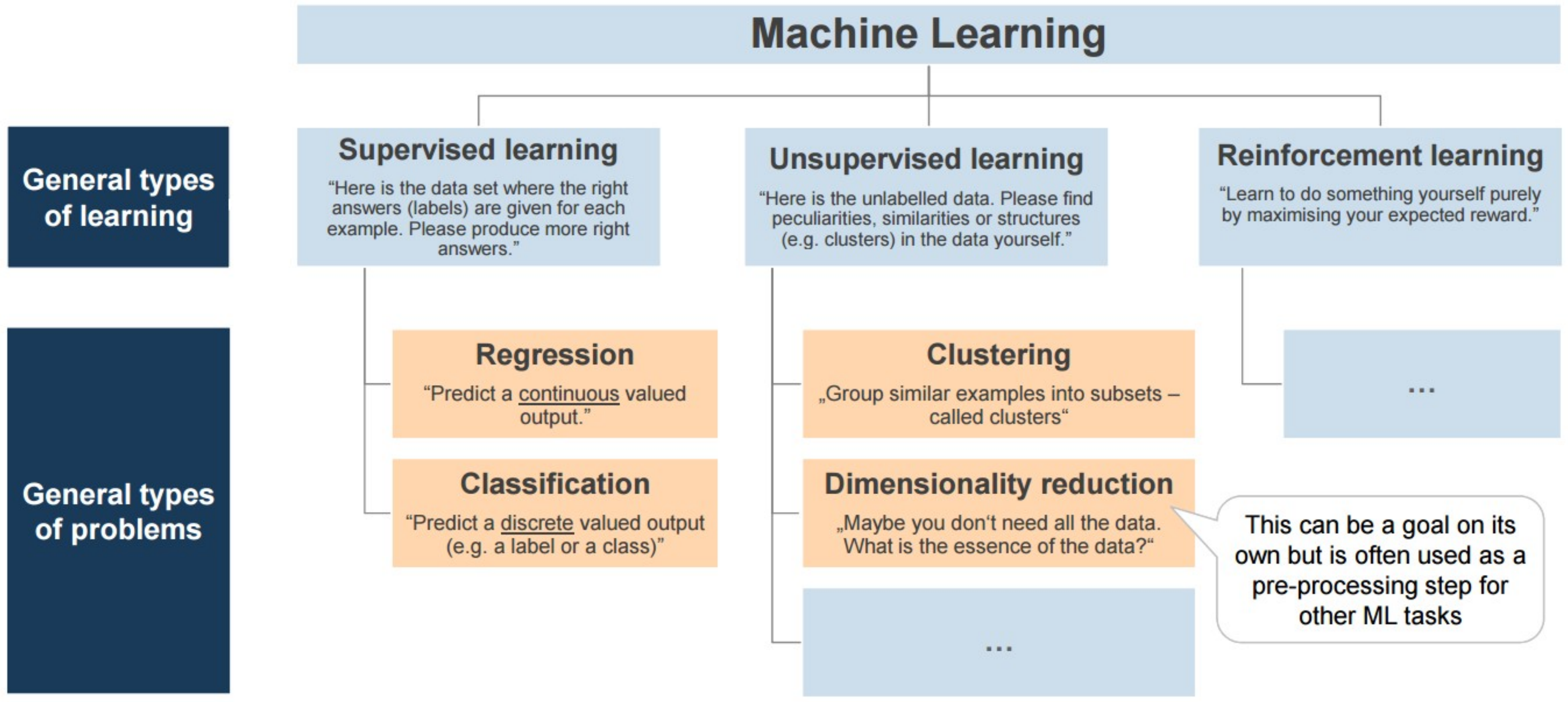
Recommendation system
("(How) will a customer like this movie?")

Identifying handwritten characters
("Which character is that supposed to be?")

Training a robot how to juggle or fly stunt manoeuvres in a helicopter



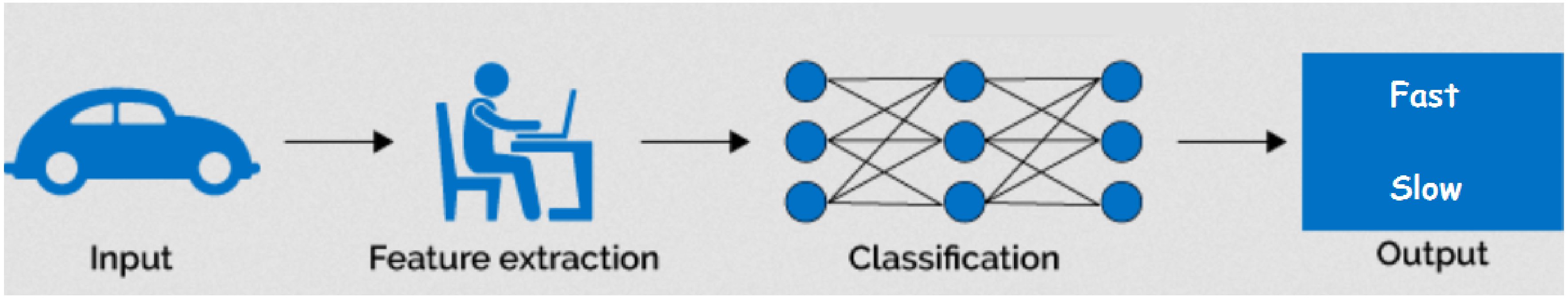
Categories of Machine Learning


Machine Learning
General types of learning
Supervised learning
“Here is the data set where the right answers (labels) are given for each example. Please produce more right answers.”
Unsupervised learning
“Here is the unlabelled data. Please find peculiarities, similarities or structures (e.g. clusters) in the data yourself.”
Reinforcement learning
“Learn to do something yourself purely by maximising your expected reward.”
General types of problems
Regression
“Predict a continuous valued output.”
Classification
“Predict a discrete valued output (e.g. a label or a class)”
Clustering
„Group similar examples into subsets – called clusters“
Dimensionality reduction
„Maybe you don‘t need all the data. What is the essence of the data?“
...
...
This can be a goal on its own but is often used as a pre-processing step for other ML tasks

Classification Problems

- Should the car go fast or slow ?
- Image Reconstruction
- Is the mushroom edible ?
- Which species the flower belongs to ? (Iris Dataset)
- Does this Pima Indian have diabetes ?
- ...

Classification: Should the car go fast or slow ?

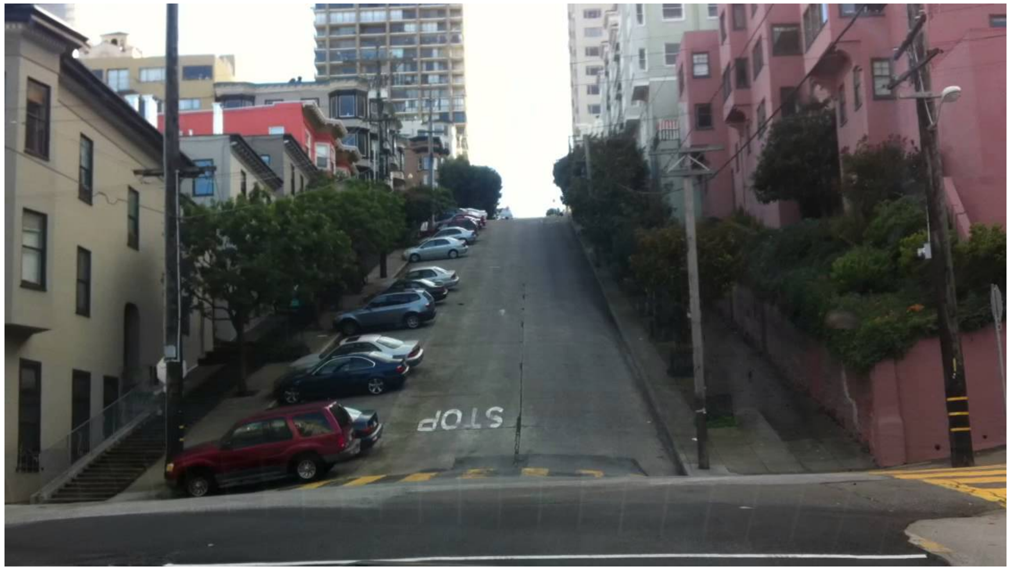


Which features ?

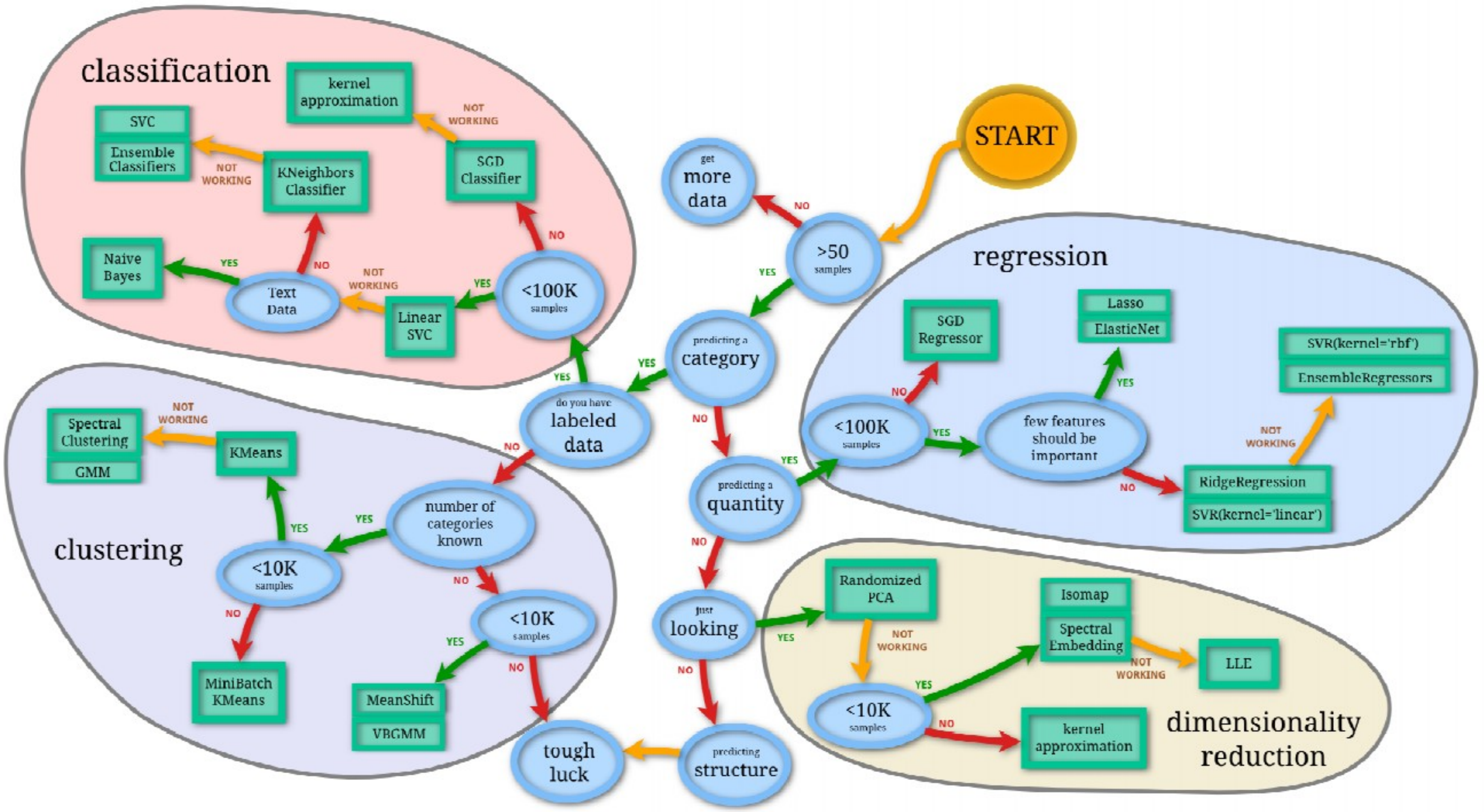
- Bumpiness

Which features ?

- Slope

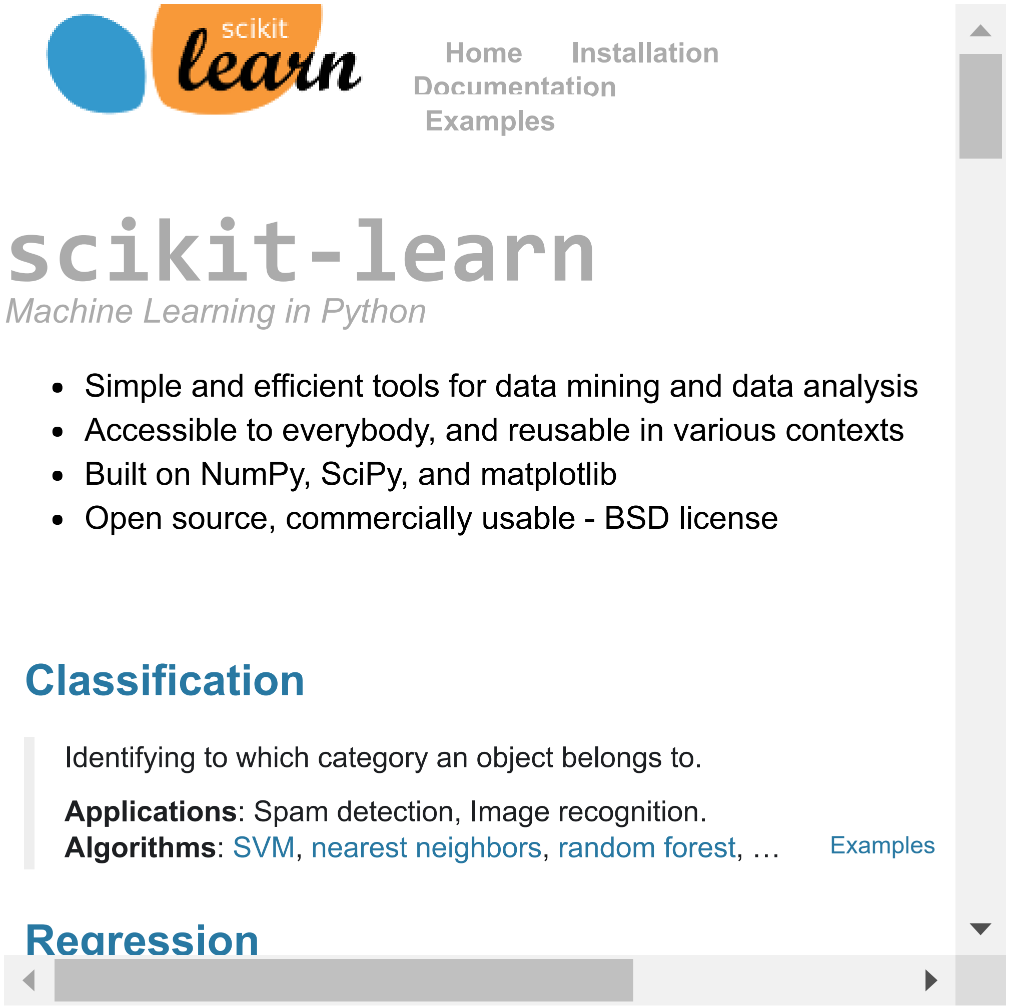


Python Sklearn Overview



Python Sklearn

http://scikit-learn.org/



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

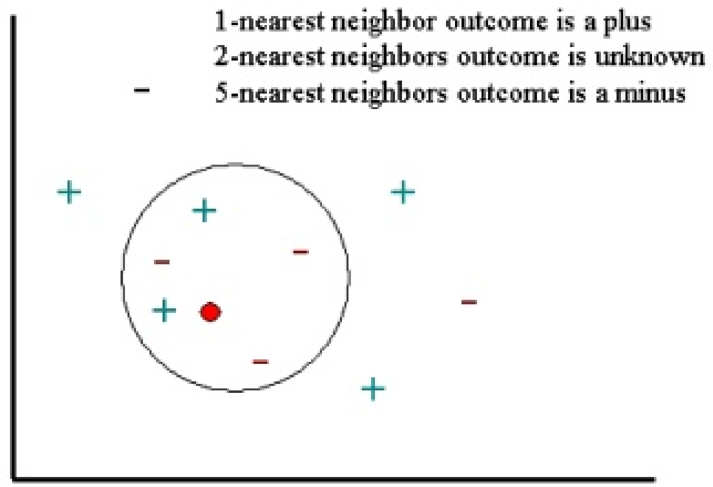
Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, …

Examples

Regression

k-nearest Neighbors

k-nearest Neighbors


1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus

k-nearest Neighbors

KNN is a non parametric lazy learning algorithm:

- It does not make any assumptions on the underlying data distribution.
- It does not use the training data points to do any generalization.

No Training Phase :) BUT it keeps all the data Warning: if it is found that two neighbors, neighbor k+1 and k, have identical distances but different labels, the results will depend on the ordering of the training data.

k-nearest Neighbors requirements

KNN assumes that the data is in a feature space: data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of **distance** (Euclidean distance or any other type)

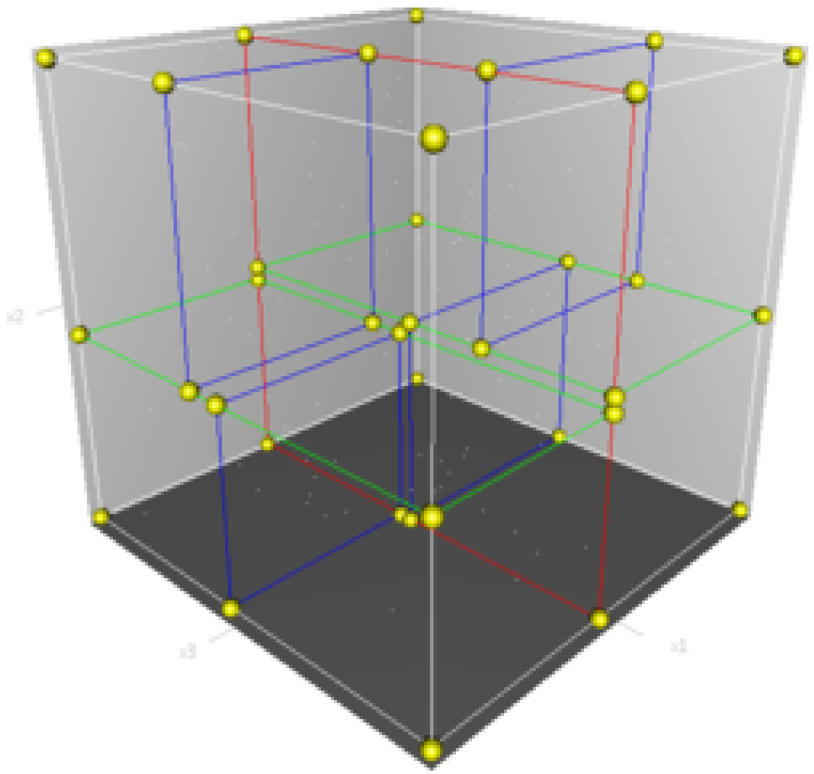
Each of the training data consists of a **set of vectors and class label** associated with each vector. In the simplest case , it will be either + or -.

We are also given a single number **k** . This number decides how many neighbors (where neighbors is defined based on the distance metric) influence the classification (usually a odd number if the number of classes is 2). If k=1 , then the algorithm is simply called the nearest neighbor algorithm.

k-nearest Neighbors Algorithms

The Algorithm used to compute the nearest neighbors could be:

- ball_tree
- kd_tree
- brute



Distance weighted k-NN algorithm

A refinement of the k-NN classification algorithm is to weigh the contribution of each of the k neighbors according to their distance to the query point xq, giving greater weight wi to closer neighbors.

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

where

$$w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

k-nearest Neighbors uses

In the case of classification, each point gets to "vote" the outcome of the observation.

In the case of regression, we take the average of the value of the points.

KNN Notebook time

Bias-variance tradeoff

The bias-variance tradeoff is the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:

- The **bias** is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (**underfitting**).
- The **variance** is error from sensitivity to small fluctuations in the training set. High variance can cause **overfitting**: modeling the random noise in the training data, rather than the intended outputs.

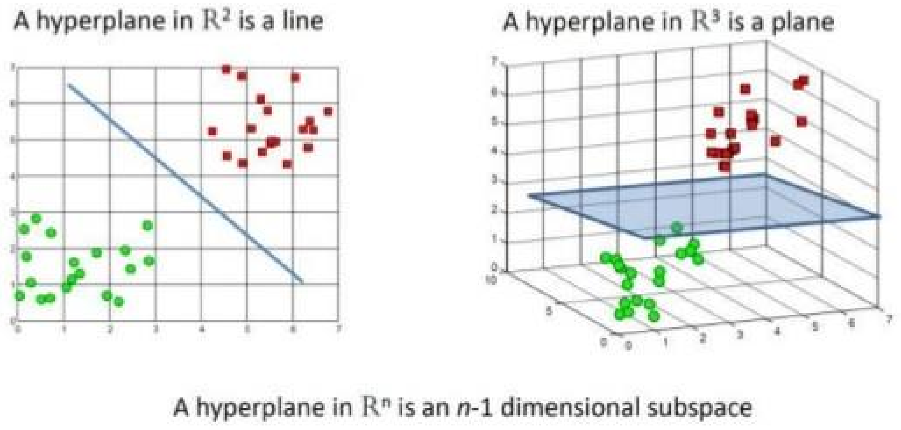
Support Vector Machines

Support Vector Machines

SVM is an approach to classification that was developed in the 90s. There are in fact 3 types of classifiers:

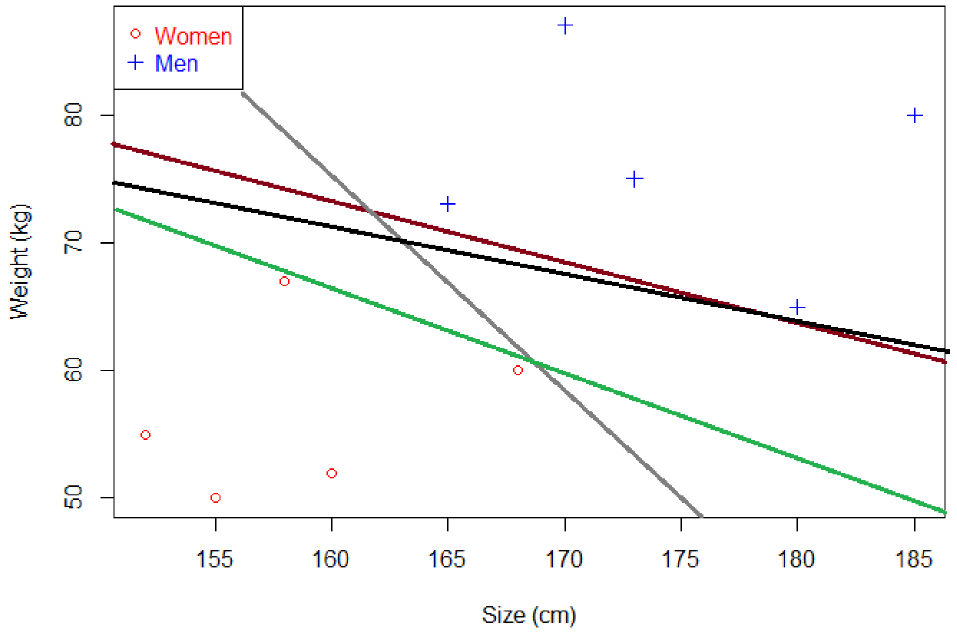
- Maximal Margin Classifier - Hard Margin
- Support Vector Classifier - Soft Margin
- Support Vector Machine - Kernel generalization

Classification



We classify the samples based on a separating hyperplane. Each point coordinates corresponds to the numerical features. Then based on the sign of f(x), we can assign each point to a half.

Which hyperplane ?



However, there are infinite separating hyperplanes. We choose the hyperplane that is the farthest from the different points. We compute the perpendicular distance from each point and we maximize that distance, also called the margin.

Hyperplane

In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension p-1.

In two dimensions, a hyperplane is defined by the equation

$$\beta 0 + \beta 1 X1 + \beta 2 X2 = 0$$

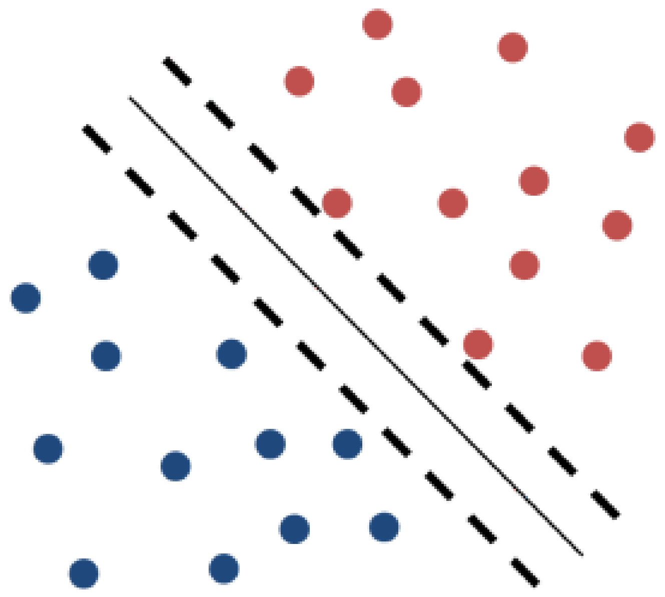
In a p-dimensional setting:

$$\beta 0 + \beta 1 X1 + \beta 2 X2 + \ldots + \beta \boldsymbol{p} X \boldsymbol{p} = 0$$

which is equivalent to:

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

Maximal Margin Classifier and Support Vectors



The observations whose points are the nearest to the hyperplane are called the support vectors. They are vectors which support the maximal margin hyperplane.

Hard Margin Optimization problem

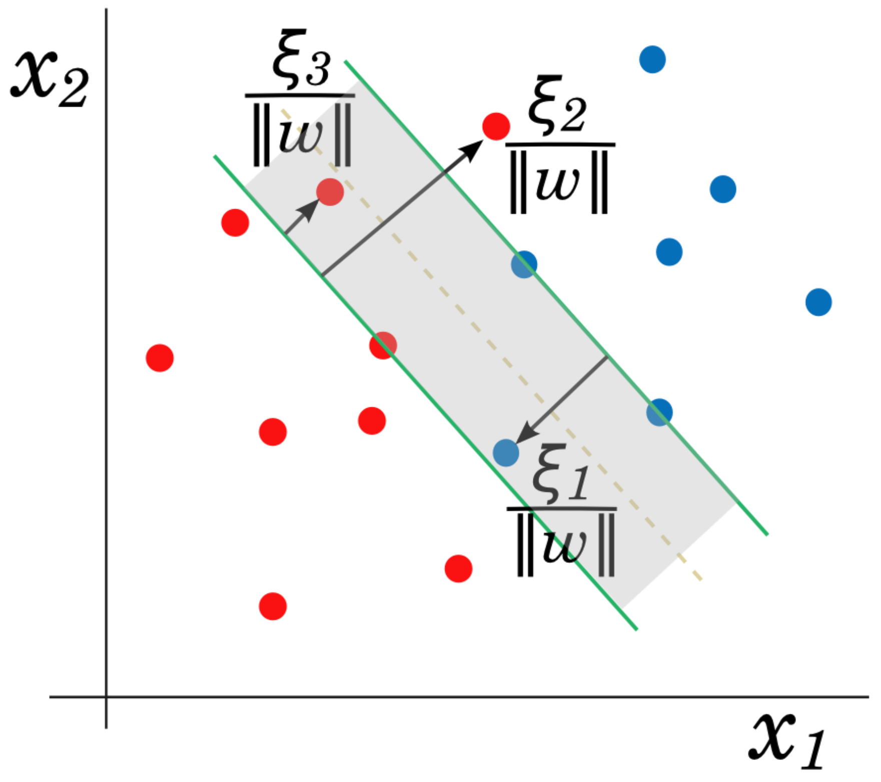
We have to solve an optimization problem to find the solutions of w:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n.$$

Soft Margin



Soft Margin Optimization problem

Our optimization problem changes slightly and we have a new parameter C:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Not linearly separable: Kernel Trick

If the observations can not be separated linearly, we can add more features to the dataset based on. However, doing this increases the computation cost a lot. Therefore we need to find a better way of computing it.

Optimization for non linear problems

The linear support vector classifier can be represented as:

$$\underset{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$

Rewriting the formula

The evaluation can be rewritten as:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle,$$

and we then can reformulate the inner product as a new non-linear function:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i).$$

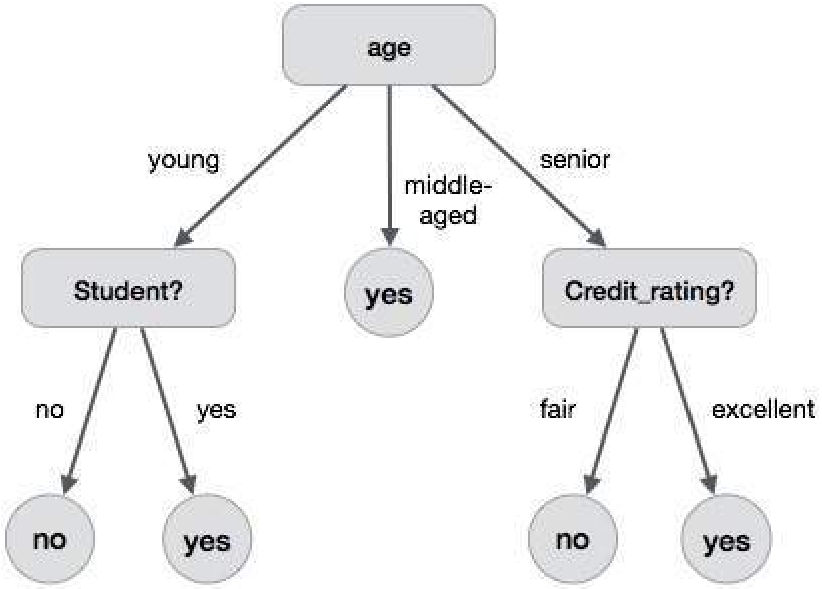
SVM Notebook Time

Decision Trees

Decision Trees

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

Example: Will it buy a computer ?



How do you build it ?

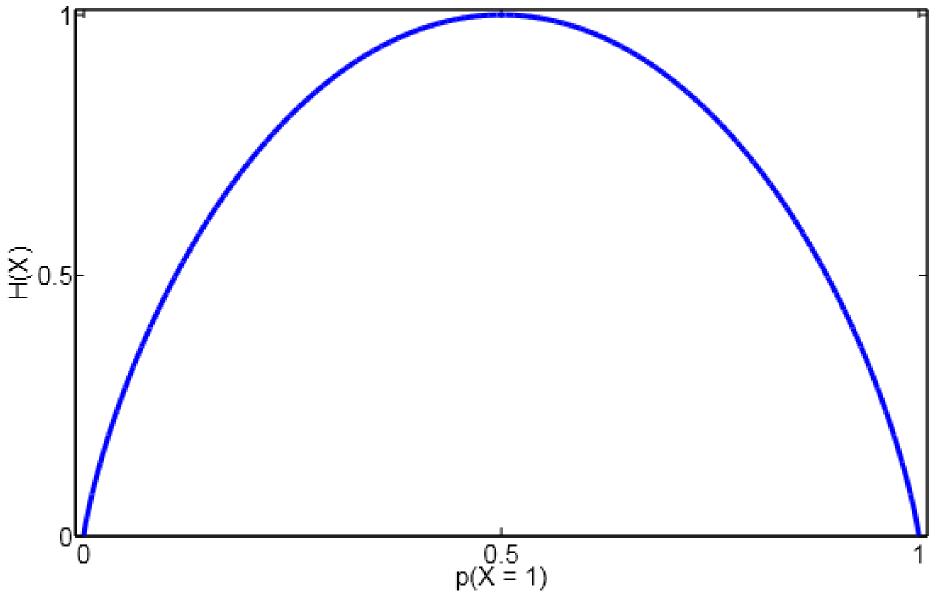
The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree.

Entropy

Entropy represents the purity of the data and can be thought as a measure of the amount of information. It is defined by the following formula:

$$H(M) = -\sum_{m \in M} p(m) \log p(m)$$

Example: Tossing a coin with a probability p for one of the faces.



Information Gain

In order to find out which feature should we select for the split, we are going to look for the split that maximizes the information gain. We define information gain as the expected reduction in entropy due to sorting on a certain field.

$$InformationGain = EntropyBefore - EntropyAfter$$

The Entropy_after is calculated from the weighted average of the entropies of the children:

$$InformationGain = Entropy(S) \; - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

DT Notebook time

Benefits of a decision tree

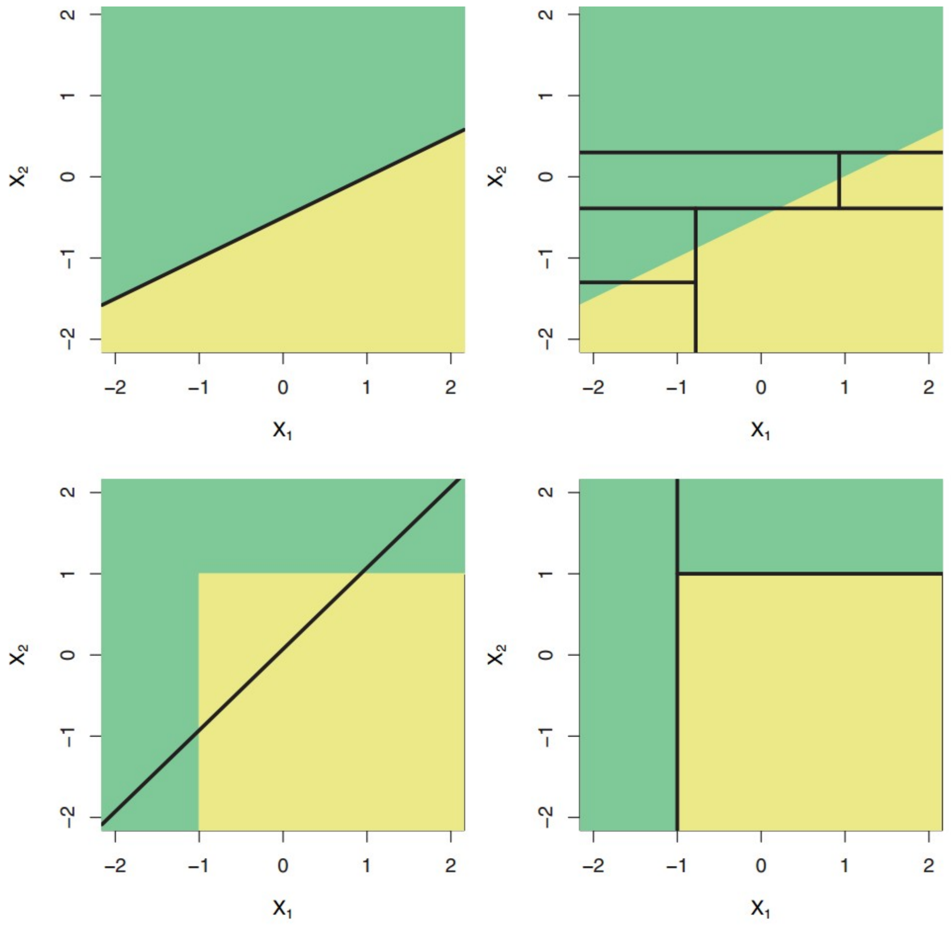
The benefits of having a decision tree are:

- It does not **require** any domain knowledge.
- It is **easy** to comprehend.
- The learning and classification steps of a decision tree are **simple and fast**.

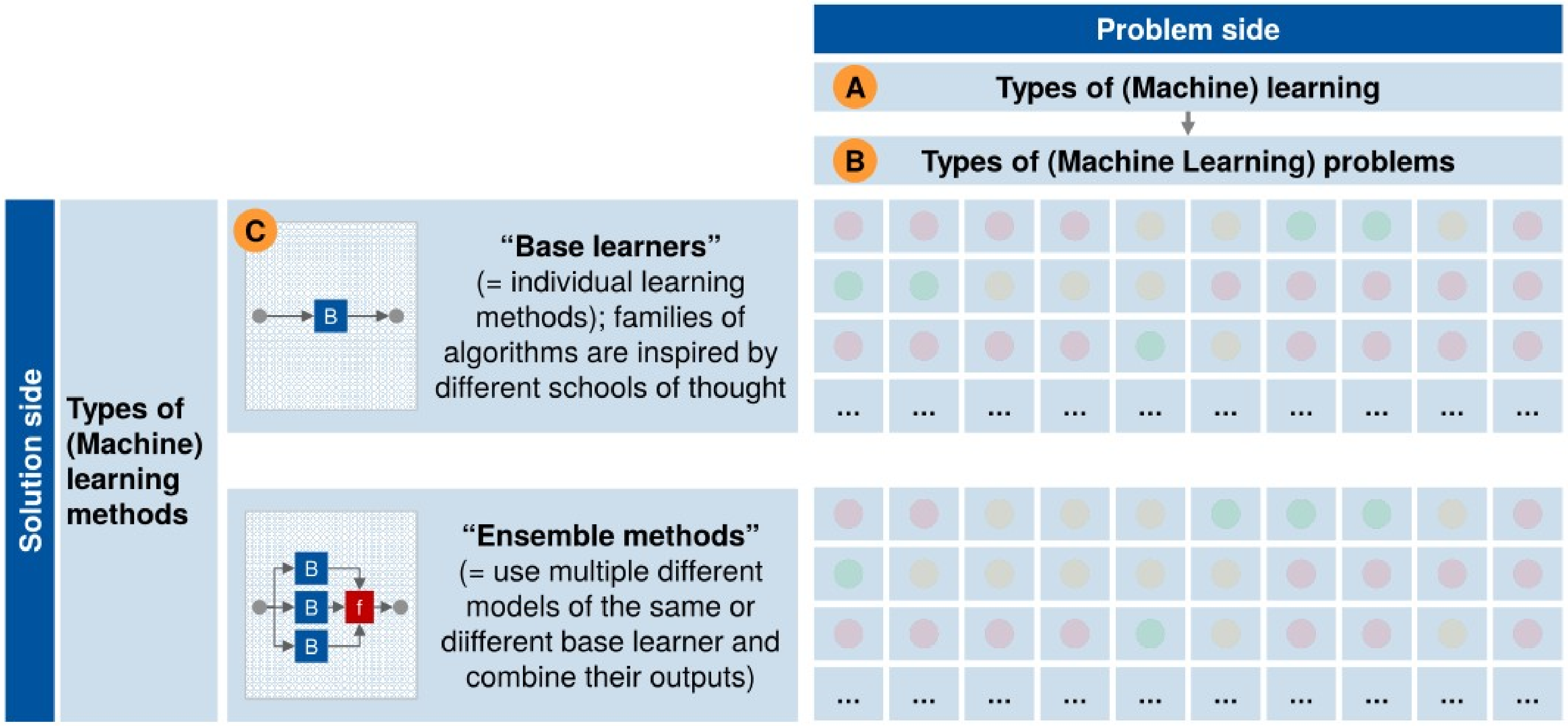
The disadvantages of decision trees:

- Unfortunately, trees generally do not have the same level of **predictive accuracy** as some of the other regression and classification approaches seen in this book.
- Additionally, trees can be very **non-robust**. In other words, a small change in the data can cause a large change in the final estimated tree.

Comparison with Linear Model



Next Steps


Problem side
A Types of (Machine) learning
B Types of (Machine Learning) problems
Solution side
Types of (Machine) learning methods
C
B
"Base learners"
(= individual learning methods); families of algorithms are inspired by different schools of thought
B
B
B
f
"Ensemble methods"
(= use multiple different models of the same or diifferent base learner and combine their outputs)
...