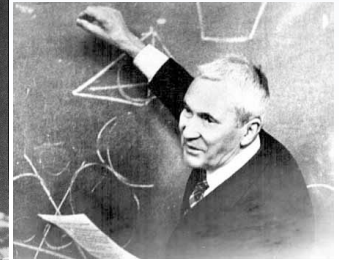
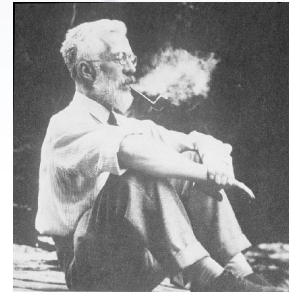
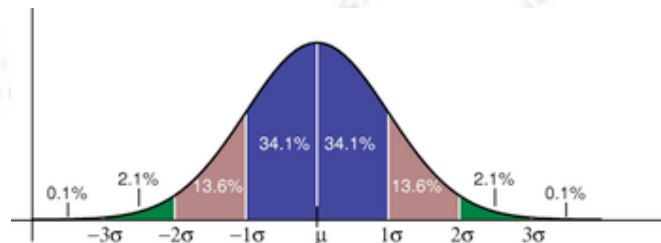


# Big Data Analysis

## Introduction to MultiVariate Analysis



Troels C. Petersen (NBI)



*“Statistics is merely a quantisation of common sense - Big Data is a sharpening of it!”*

# Dimensionality and Complexity

Humans are good at seeing/understanding data in few dimensions!

However, as dimensionality grows, complexity grows exponentially (“curse of dimensionality”), and humans are generally not geared for such challenges.

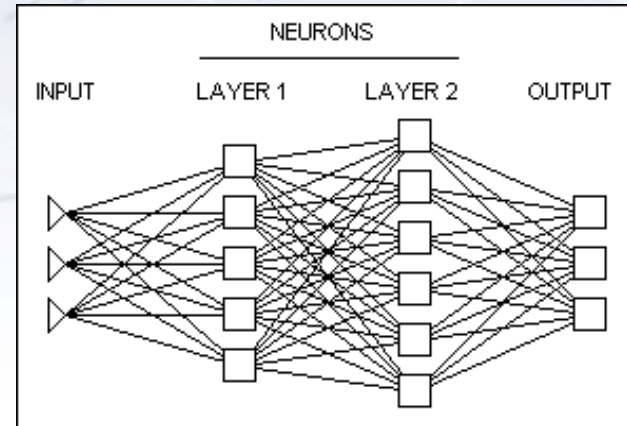
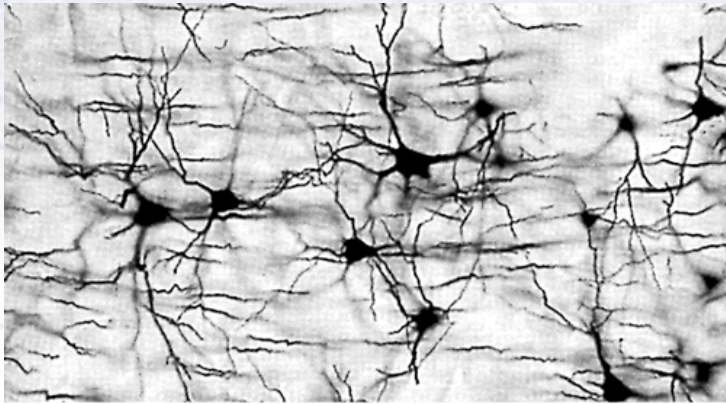
	Low dim.	High dim.
Linear	Humans: ✓ Computers: ✓	Humans: ÷ Computers: ✓
Non-linear	Humans: ✓ Computers: (✓)	Humans: ÷ Computers: (✓)

Computers, on the other hand, are OK with high dimensionality, albeit the growth of the challenge, but have a harder time facing non-linear issues.

However, through smart algorithms, computers have learned to deal with it all!

# Data Mining

Seeing patterns in data and using it!



*Data mining is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and **scientific discovery**.*

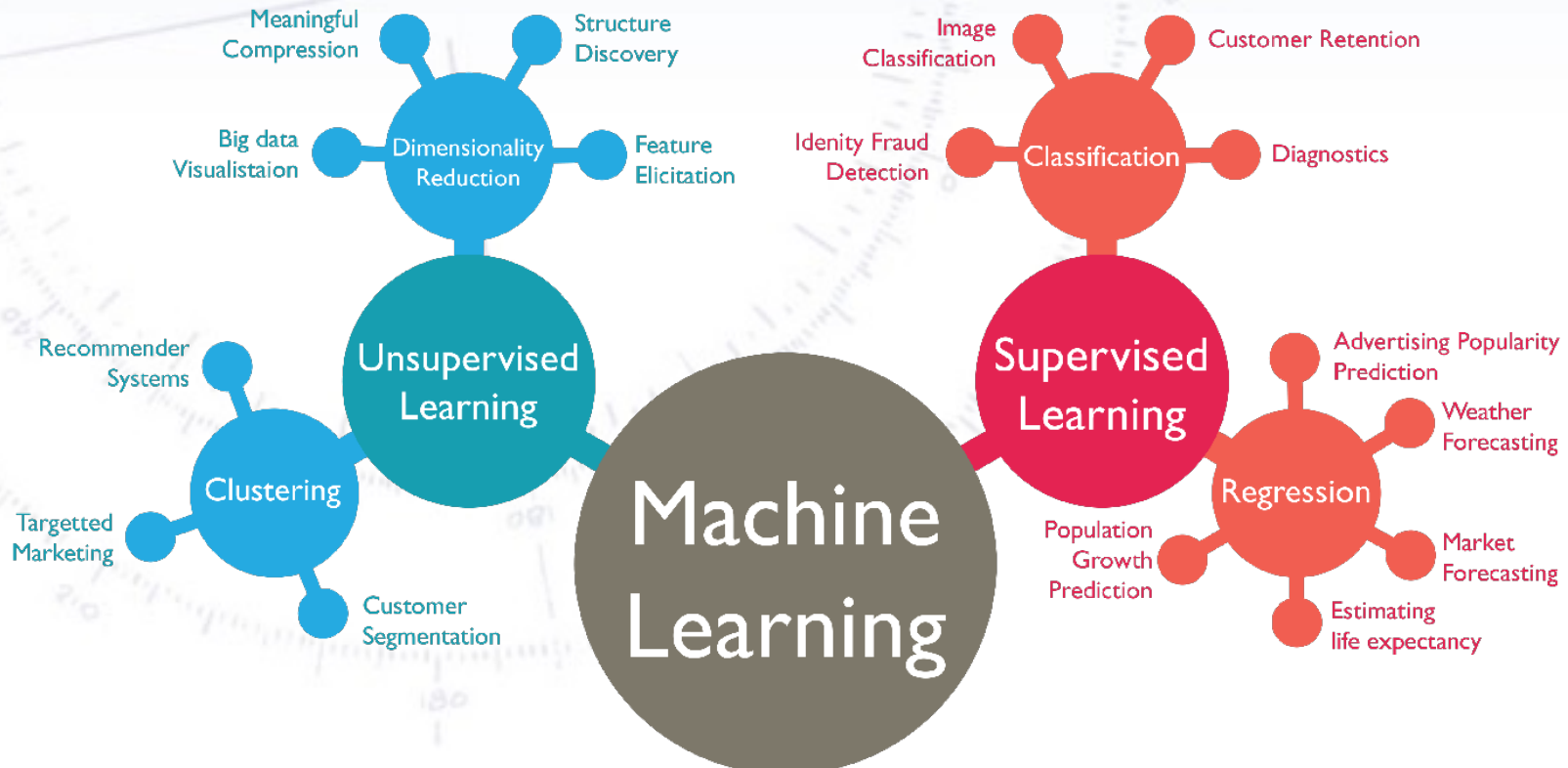
[Wikipedia, Introduction to Data Mining]



# Classification vs. Regression

## Unsupervised learning vs. supervised

Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)... [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).

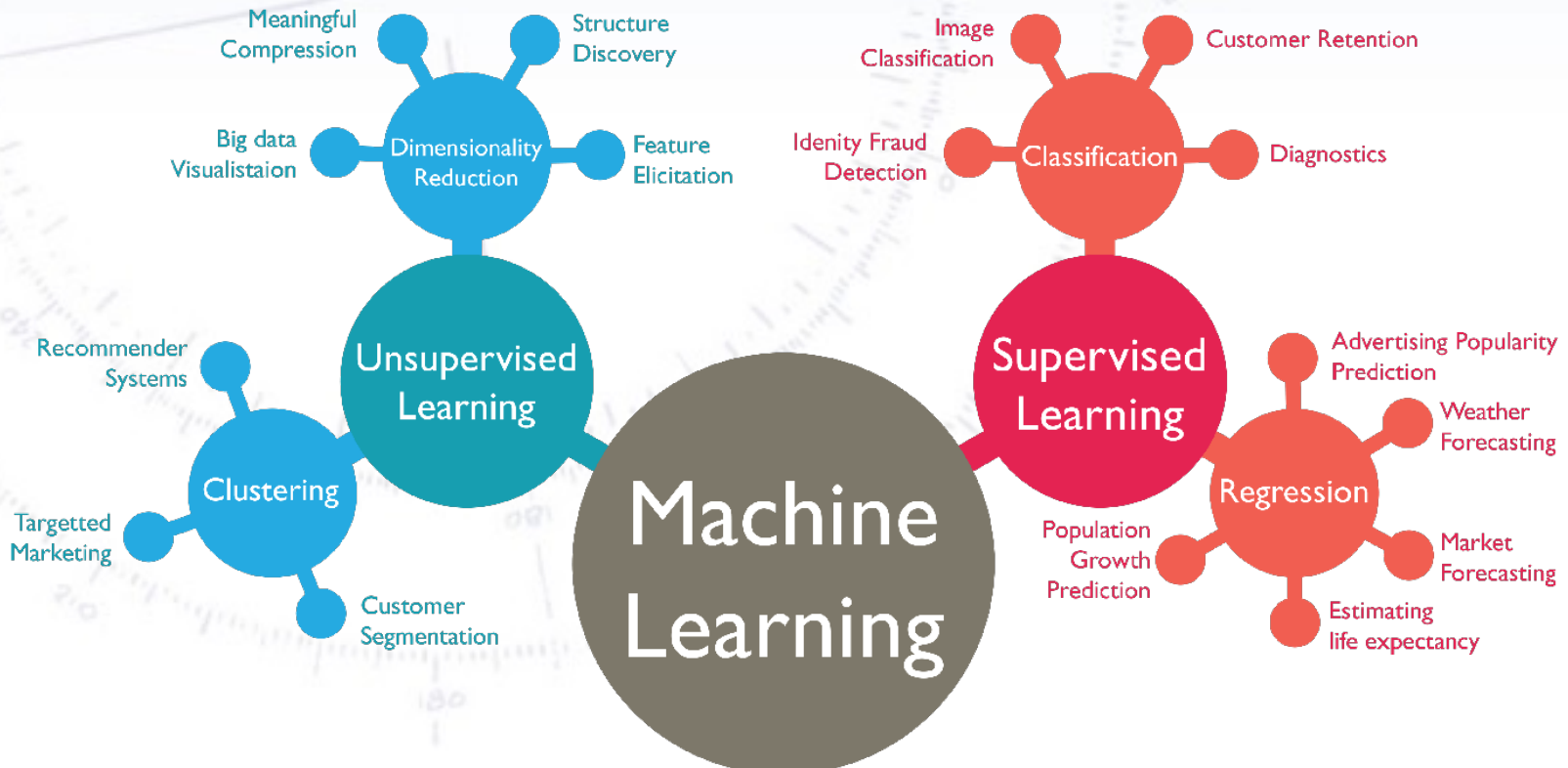


# Classification vs. Regression

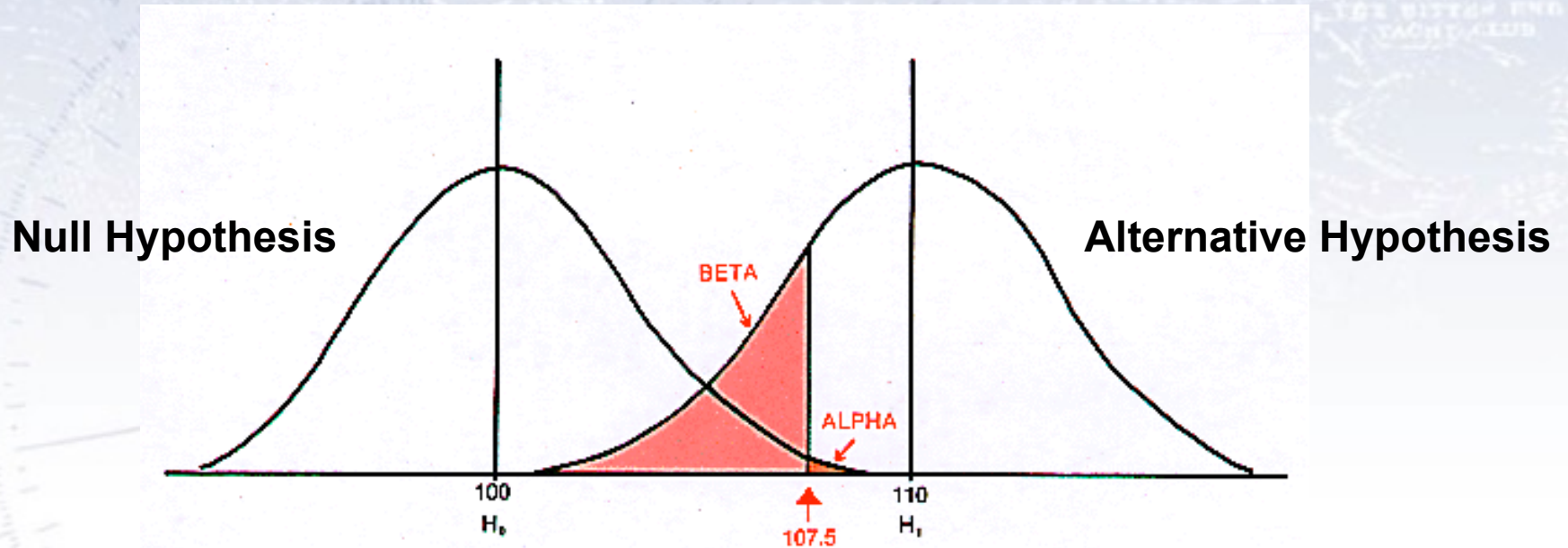
## Unsupervised learning vs. supervised

Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)... [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).

**We will be mostly on this side!**

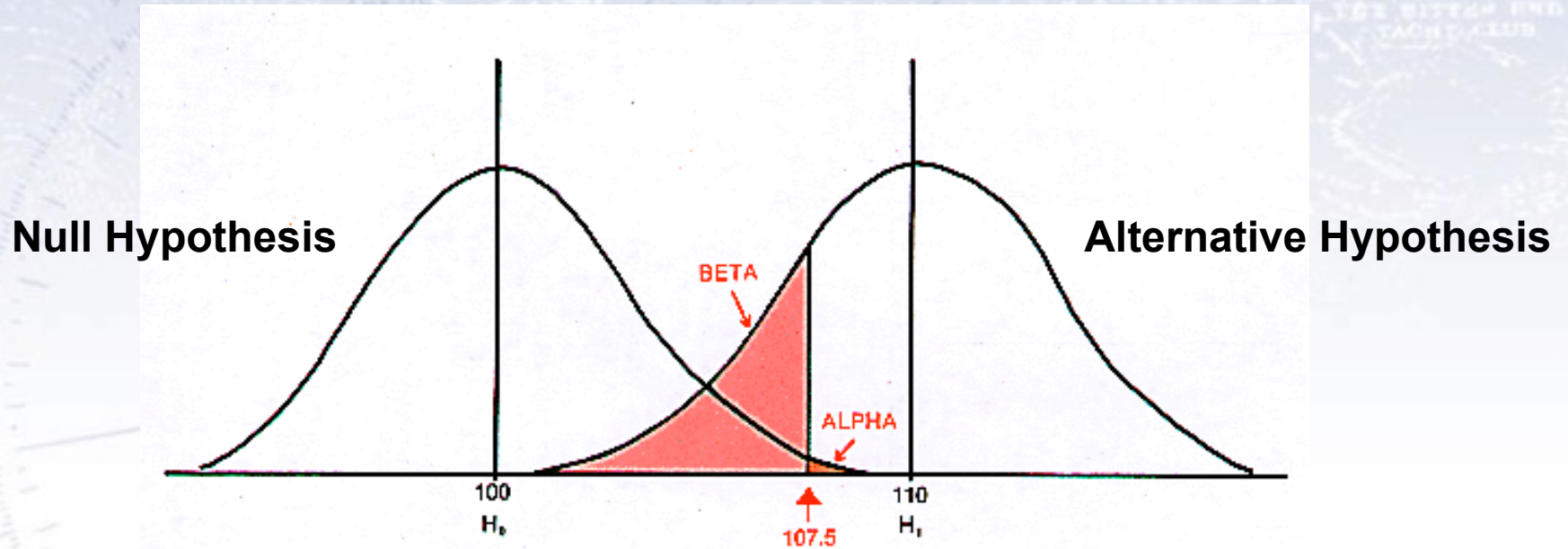


# Classification/Hypothesis



		REALITY	
		Null is True	Null is False
STATISTICAL DECISION:	Do Not Reject Null	$1 - \alpha$ Correct	$\beta$ Type II error
	Reject Null	$\alpha$ Type I error	$1 - \beta$ Correct

# Classification/Hypothesis



Machine Learning typically enables a better separation between hypothesis

**DECISION:**

Reject Null

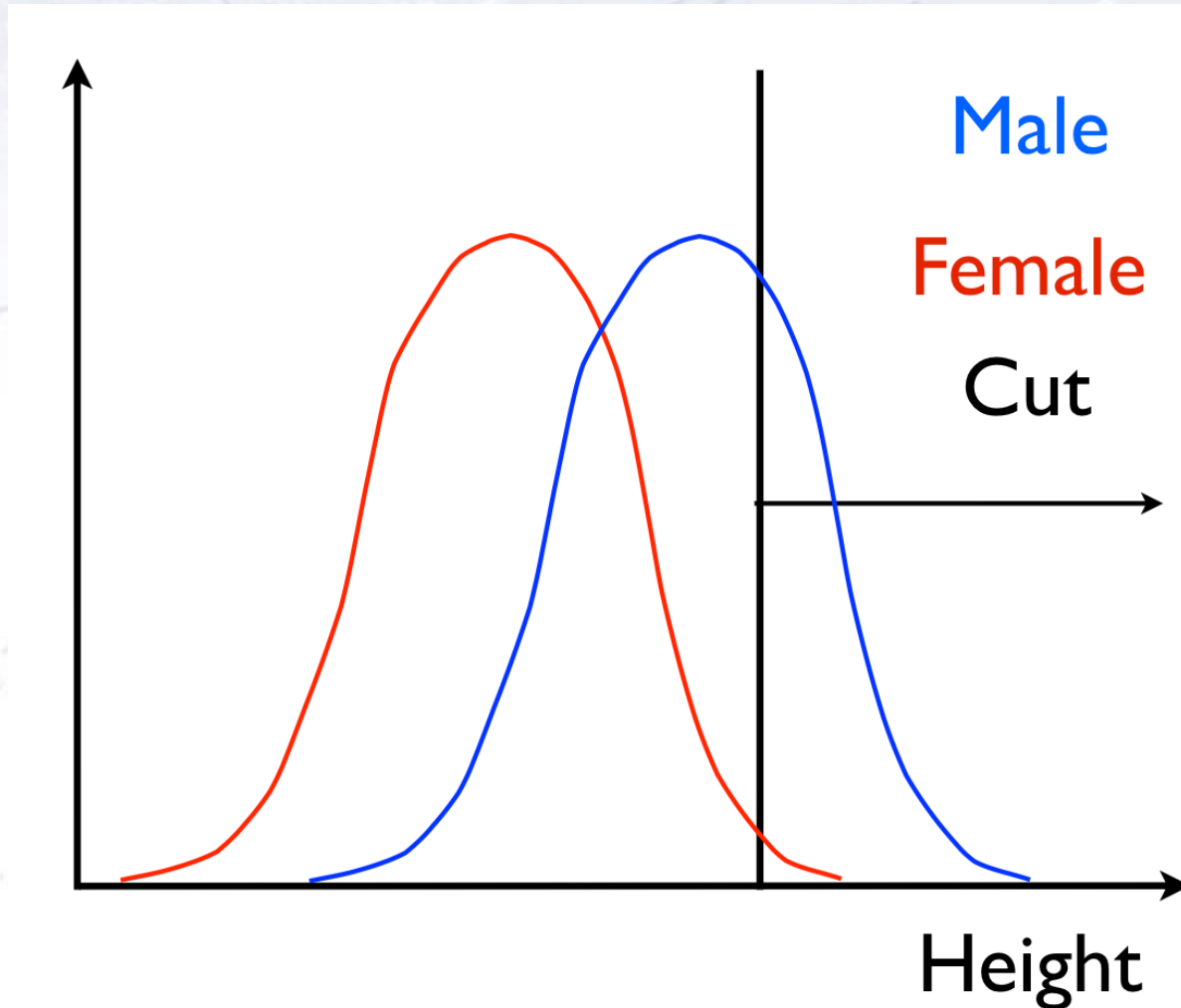
$\alpha$ Type I error	$1 - \beta$ Correct
--------------------------	------------------------



# Simple Example

**Problem:** You want to figure out a method for getting sample that is 95% male!

**Solution:** Gather height data from 10000 people, Estimate cut with 95% purity!

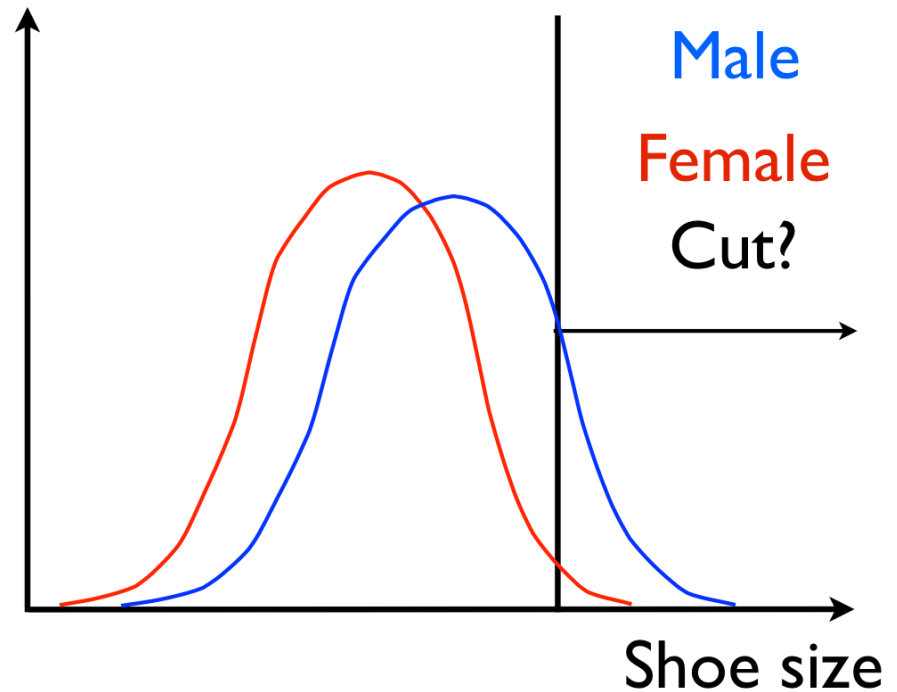
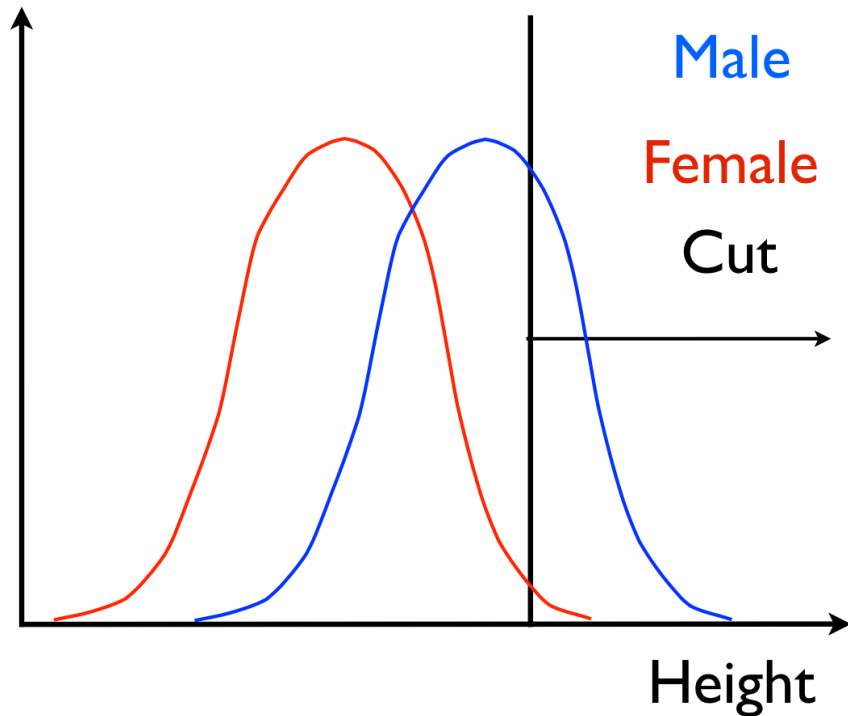




# Simple Example

**Additional data:** The data you find also contains shoe size!

**How to use this?** Well, it is more information, but should you cut on it?



The question is, what is the best way to use this (possibly correlated) information!

# Simple Example

So we look if the data is correlated, and consider the options:

Cut on each var?

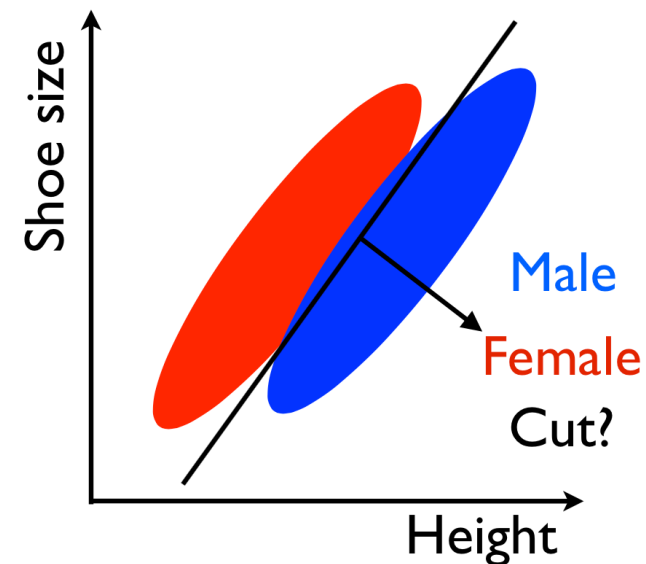
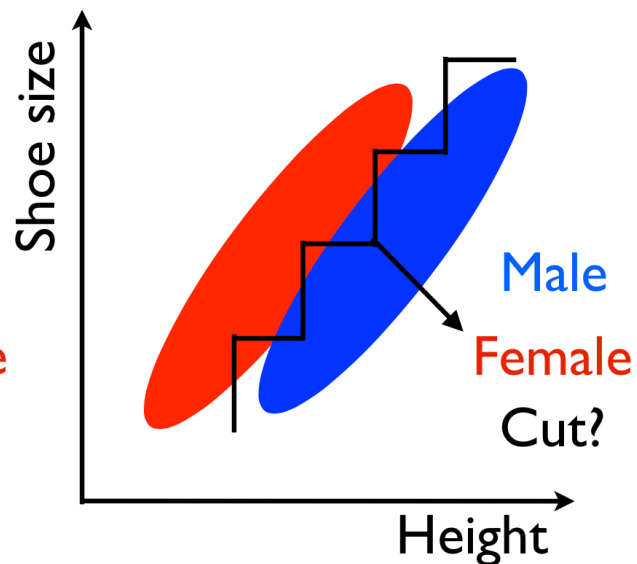
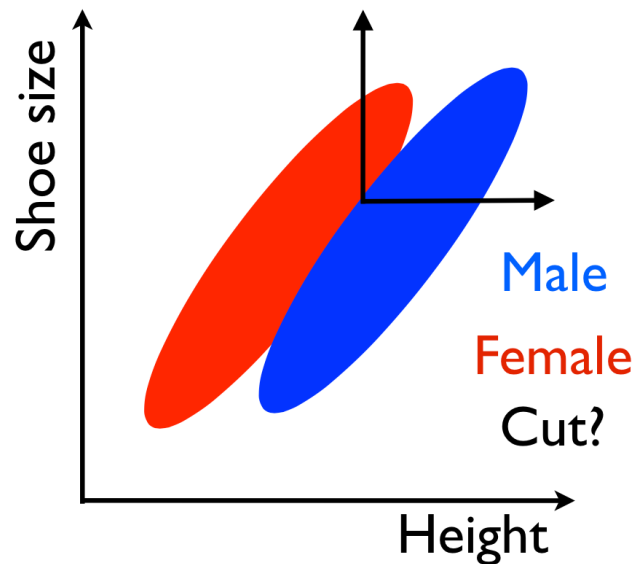
**Poor efficiency!**

Advanced cut?

**Clumsy and  
hard to implement**

Combine var?

**Smart and  
promising**



The latter approach is the Fisher discriminant!




It has the advantage of being simple and applicable in many dimensions easily!

# Separating data

Fisher's friend, Anderson, came home from picking Irises in the Gaspé peninsula...

## 180 MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS

Table I

<i>Iris setosa</i>				<i>Iris versicolor</i>				<i>Iris virginica</i>			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
											
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3



# Fisher's Linear Discriminant

You want to separate two types/classes (A and B) of events using several measurements.

**Q:** How to combine the variables?

**A:** Use the Fisher Discriminant:

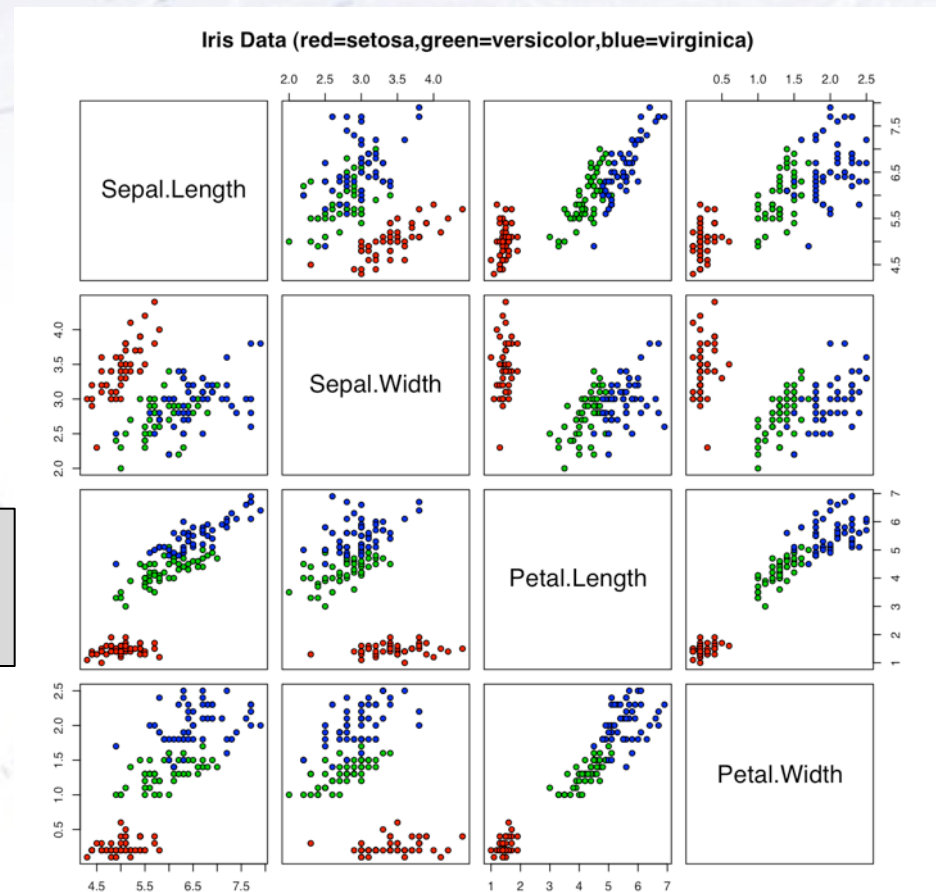
$$\mathcal{F} = w_0 + \vec{w} \cdot \vec{x}$$

**Q:** How to choose the values of  $w$ ?

**A:** Inverting the covariance matrices:

$$\vec{w} = (\Sigma_A + \Sigma_B)^{-1} (\vec{\mu}_A - \vec{\mu}_B)$$

This can be calculated analytically, and incorporates the linear correlations into the separation capability.





# Fisher Discriminant

You want to separate two types/classes (A and B) of events using several measurements.

**Q:** How to combine the variables?

**A:** Use the Fisher Discriminant:

measurements are given. We shall first consider the question: What linear function of the four measurements

$$X = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4 x_4$$

will maximize the ratio of the difference between the specific means to the standard deviations within species? The observed means and their differences are shown in Table II.

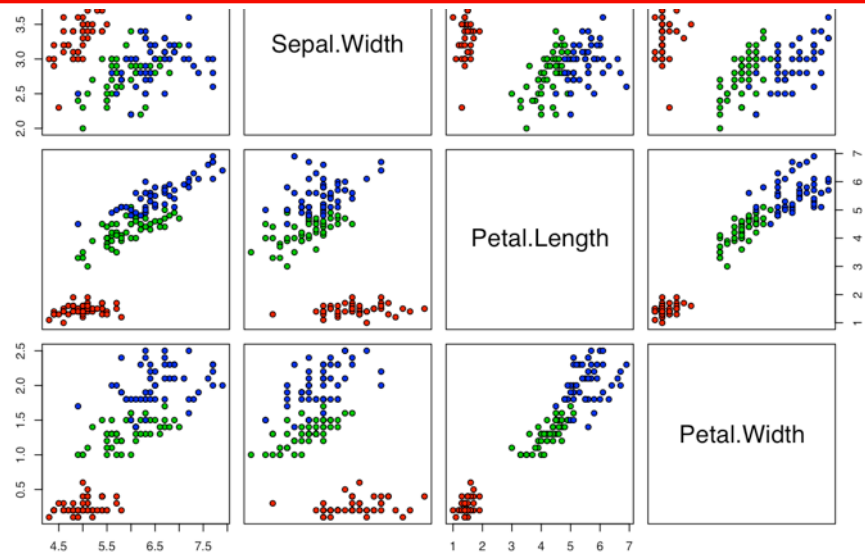
**Q:** How to choose the values of  $w$ ?

**A:** Inverting the covariance matrices:

$$\vec{w} = (\Sigma_A + \Sigma_B)^{-1} (\vec{\mu}_A - \vec{\mu}_B)$$

This can be calculated analytically, and incorporates the linear correlations into the separation capability.

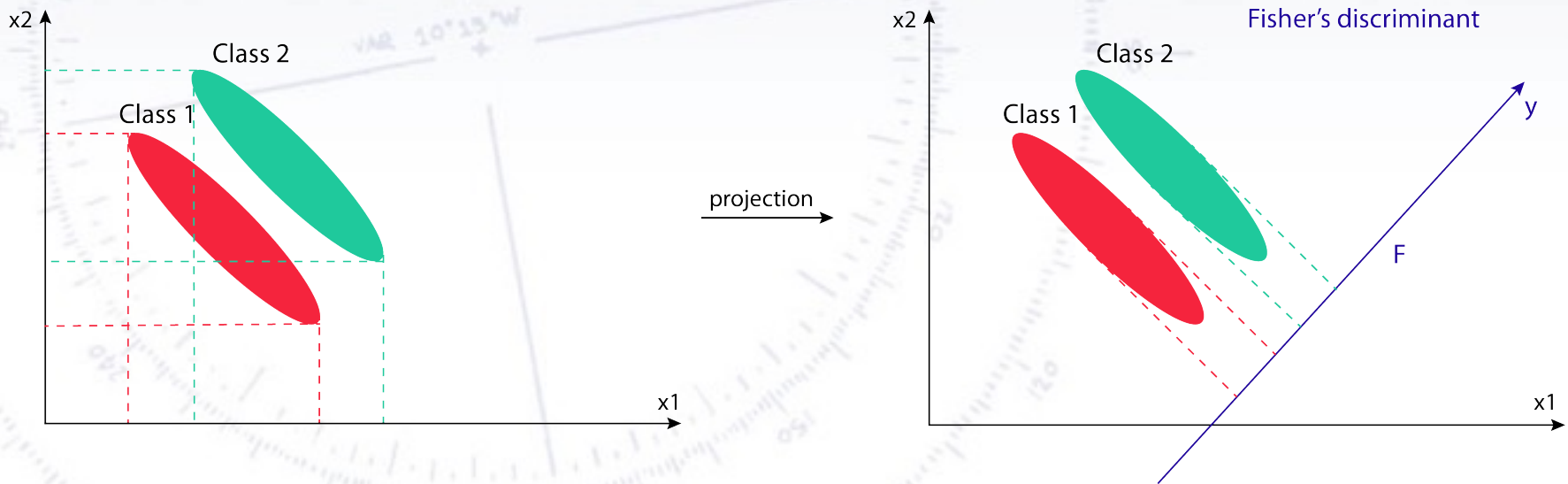
Iris Data (red=setosa,green=versicolor,blue=virginica)



# Fisher Discriminant

## Executive summary:

Fisher's Discriminant uses a linear combination of variables to give a single variable with the maximum possible separation (for linear combinations!).



It is for all practical purposes a projection (in a Euclidian space)!

# Fisher Discriminant

The details of the formula are outlined below:

You have two samples, A and B, that you want to separate.

For each input variable (x), you calculate the mean ( $\mu$ ), and form a vector of these.

$$\vec{w} = (\Sigma_A + \Sigma_B)^{-1} (\vec{\mu}_A - \vec{\mu}_B)$$

Using the input variables (x), you calculate the covariance matrix ( $\Sigma$ ) for each species (A/B), add these and invert.

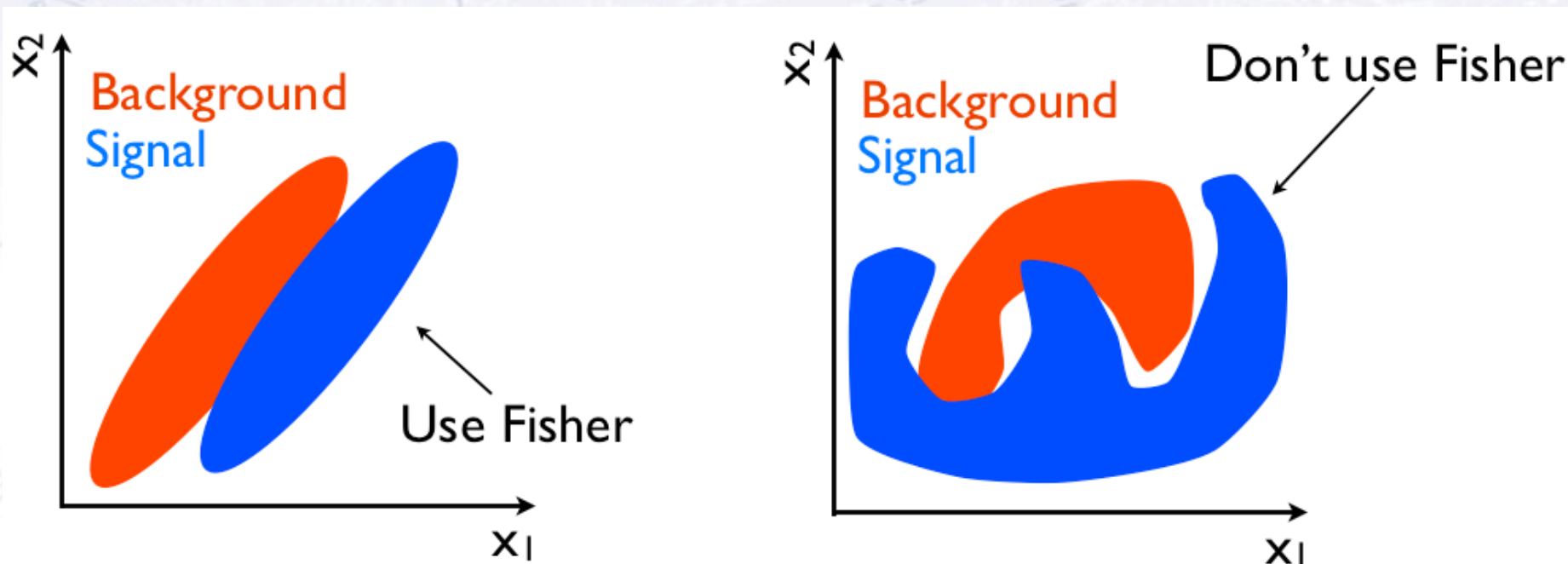
Given weights (w), you take your input variables (x) and combine them linearly as follows:

$$\mathcal{F} = w_0 + \vec{w} \cdot \vec{x}$$

F is what you base your decision on.

# Non-linear MVAs

While the Fisher Discriminant uses all separations and **linear correlations**, it does not perform optimally, when there are **non-linear correlations** present:



If the PDFs of signal and background are known, then one can use a likelihood. But this is **very rarely** the case, and hence one should move on to the Fisher. However, if correlations are non-linear, more “tough” methods are needed...



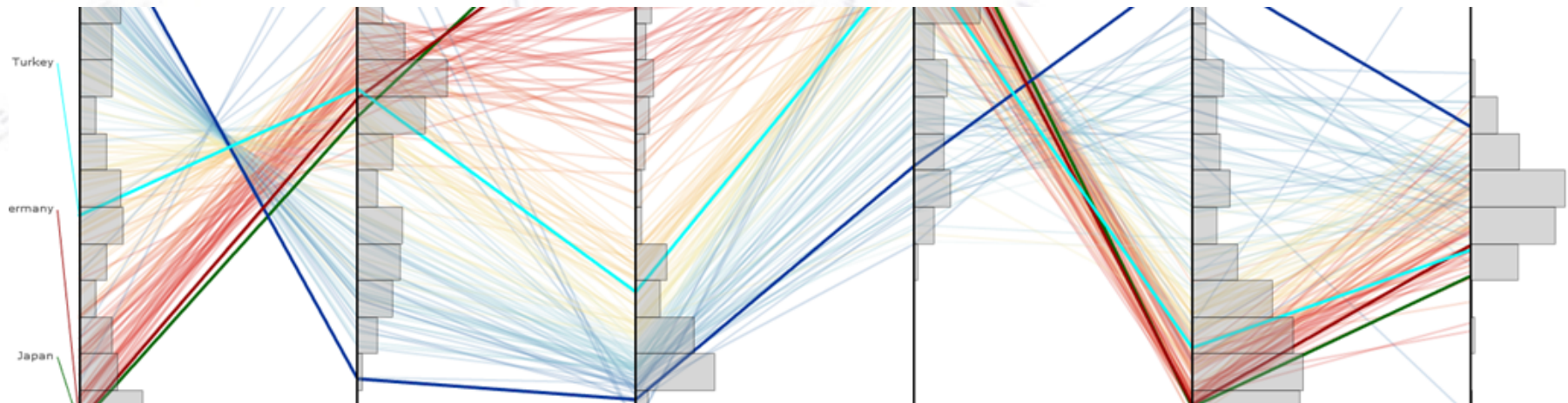
# Today's goal: Introduction

MultiVariate Analysis (MVA) is a **huge subject**, and it is **impossible** to go into any detail in one day.

The goal of today's exercise is to:

- Give you an introduction to more advanced MVA methods.
- Be able to recognise problems, where MVA is applicable.
- Wet your appetite for advanced MVA methods.

So let us dive into the world of extracting knowledge from information.



# Neural Networks

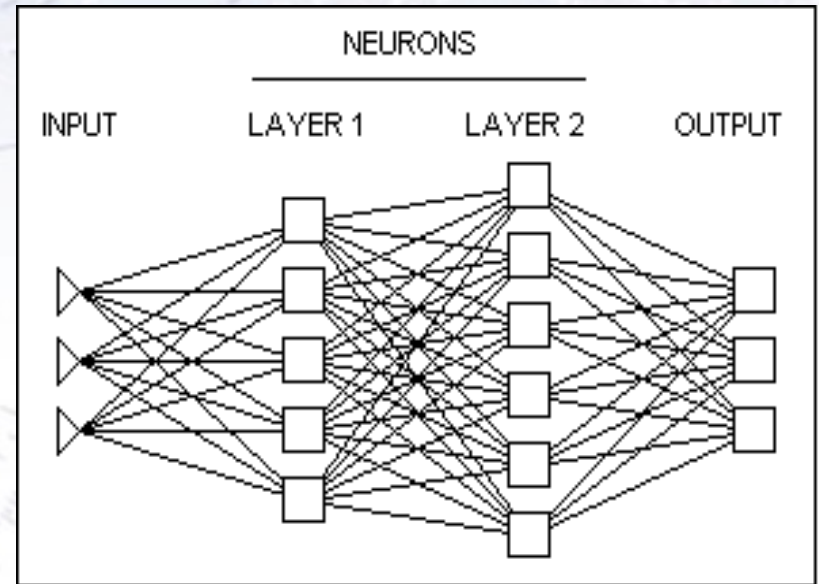
Can become very complex.

Good for continuous problems.

Sometimes hard to train!

Can be used for images.

Easily produces multiple outputs.



# (Boosted) Decision Trees

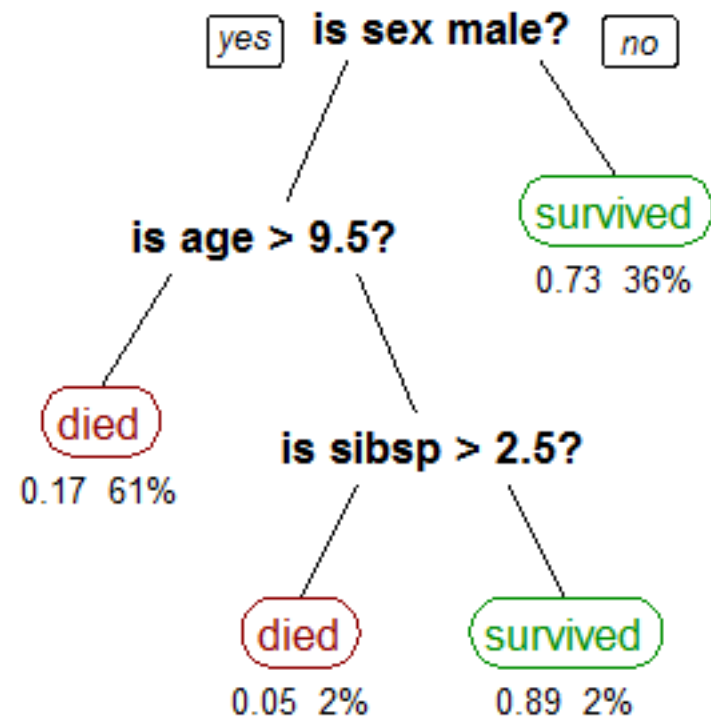
Can become very complex.

Good for discrete problems.

“Good for all problems!!!”

Not always highest efficiency.

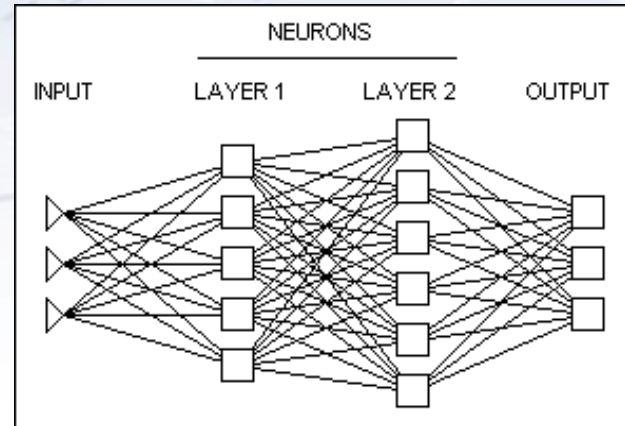
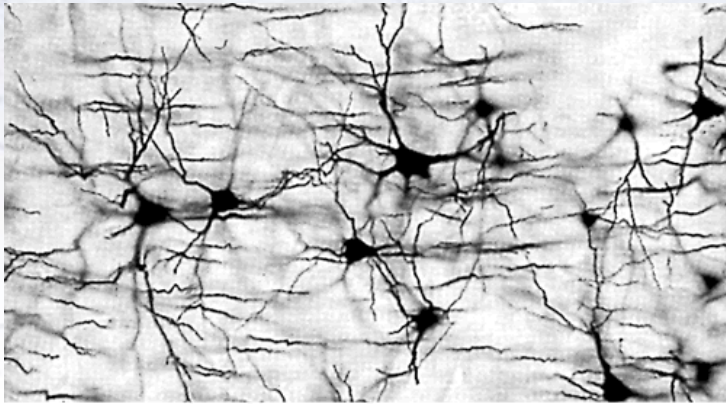
Boosting adds to separation.



\* The example BDT shown is a simple example for predicting survival of Titanic!



# Neural Networks (NN)



*In machine learning and related fields, artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems (in particular the brain) which is capable of **machine learning** as well as **pattern recognition**.*

*Neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including **computer vision** and **speech recognition**.*

[Wikipedia, Introduction to Artificial Neural Network]



# Neural Networks

Neural Networks combine the input variables using a “activation” function  $s(x)$  to assign, if the variable indicates signal or background.

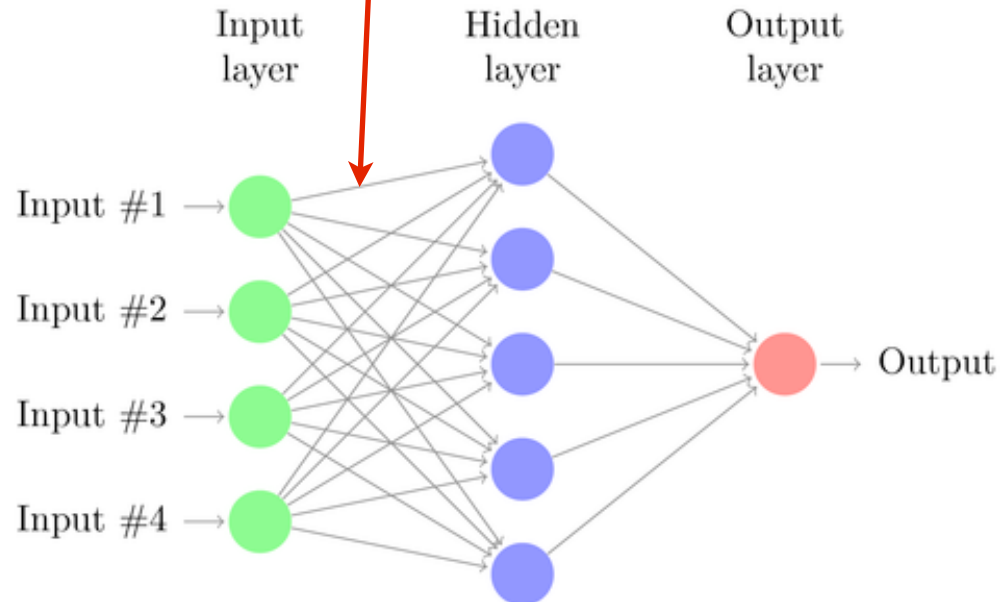
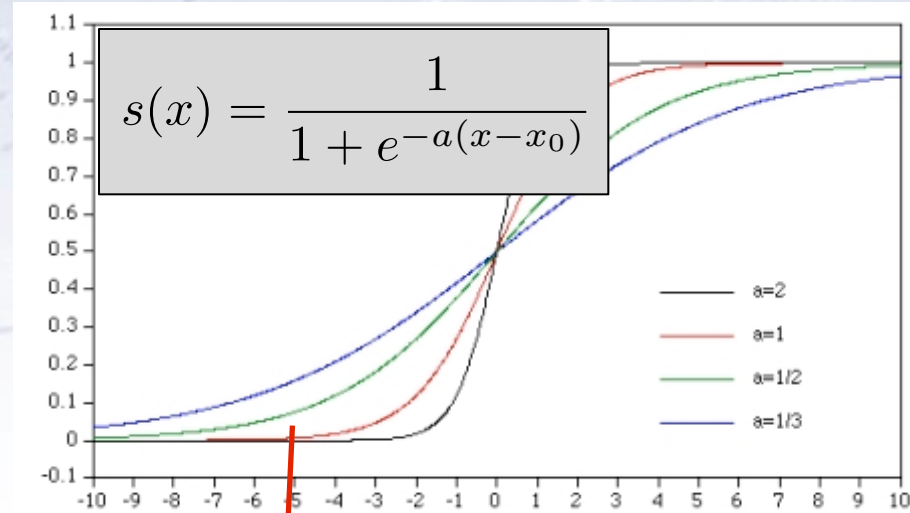
The simplest is a single layer perceptron:

$$t(x) = s \left( a_0 + \sum a_i x_i \right)$$

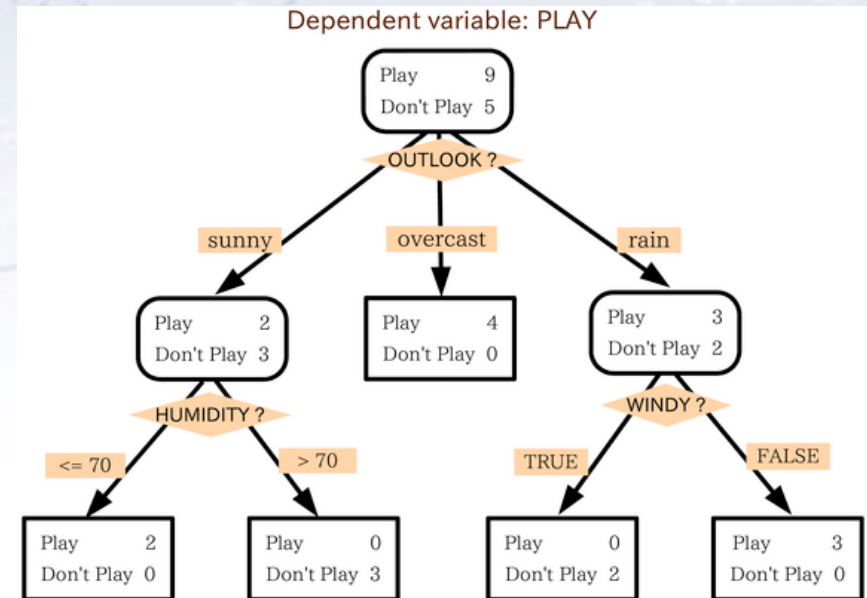
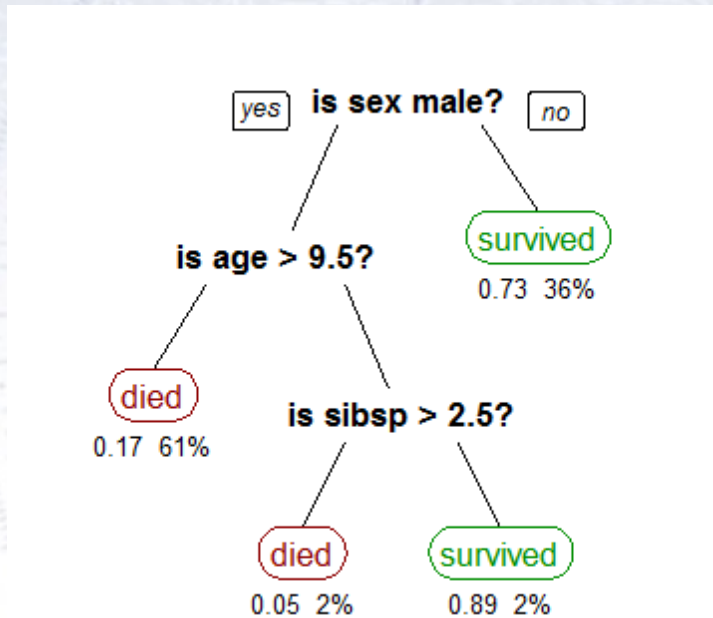
This can be generalised to a multilayer perceptron:

$$t(x) = s \left( a_i + \sum a_i h_i(x) \right)$$
$$h_i(x) = s \left( w_{i0} + \sum w_{ij} x_j \right)$$

Activation function can be any sigmoid function.



# Boosted Decision Trees (BDT)



*Decision tree learning uses a **decision tree** as a **predictive model** which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in **statistics**, **data mining** and **machine learning**.*

[Wikipedia, Introduction to Decision Tree Learning]

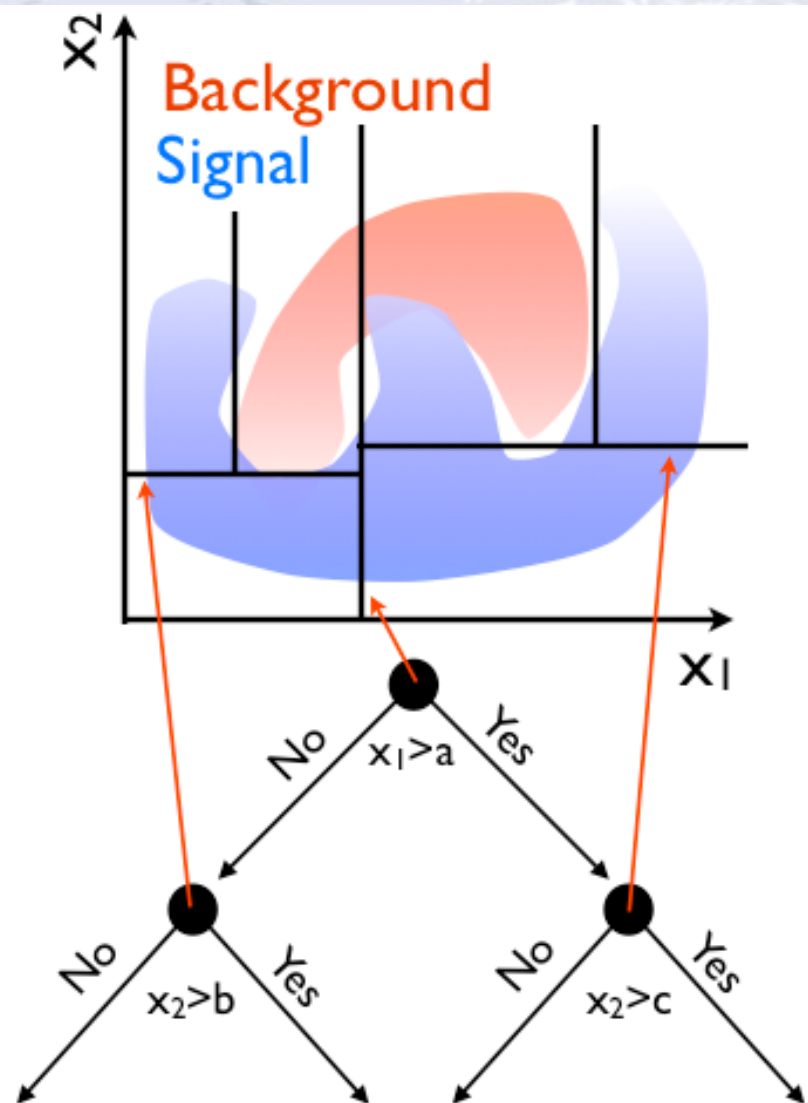
# Boosted Decision Trees

A decision tree divides the parameter space, starting with the maximal separation. In the end each part has a probability of being signal or background.

- Works in 95+% of all problems!
- Fully uses non-linear correlations.

But BDTs require a lot of data for training, and is sensitive to overtraining (see next slide).

Overtraining can be reduced by limiting the number of nodes and number of trees.



# Boosting...

There is no reason, why you can not have more trees. Each tree is a simple classifier, but many can be combined!

To avoid N identical trees, one assigns a higher weight to events that are hard to classify, i.e. boosting:

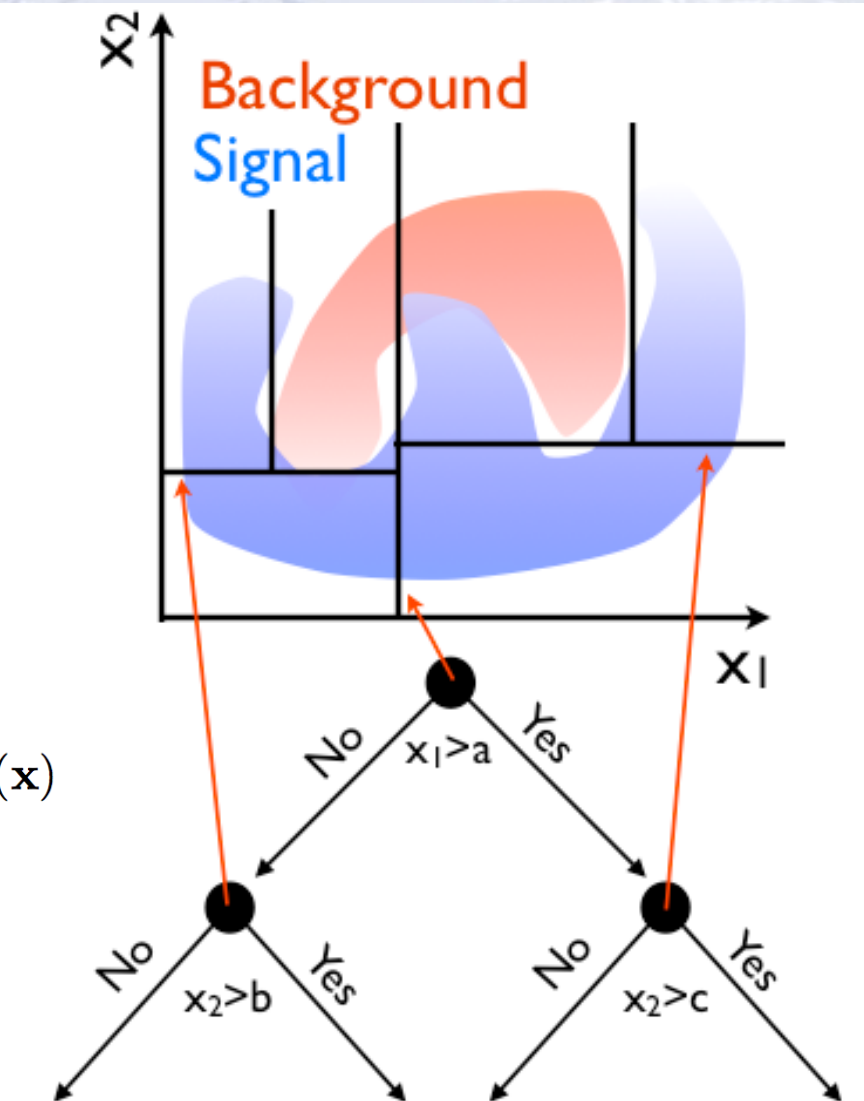
First classifier

$$y_{\text{Boost}}(\mathbf{x}) = \frac{1}{N_{\text{collection}}} \cdot \sum_i^{N_{\text{collection}}} \ln(\alpha_i) \cdot h_i(\mathbf{x})$$

Parameters in event N

Boost weight  $\alpha = \frac{1 - \text{err}}{\text{err}}$

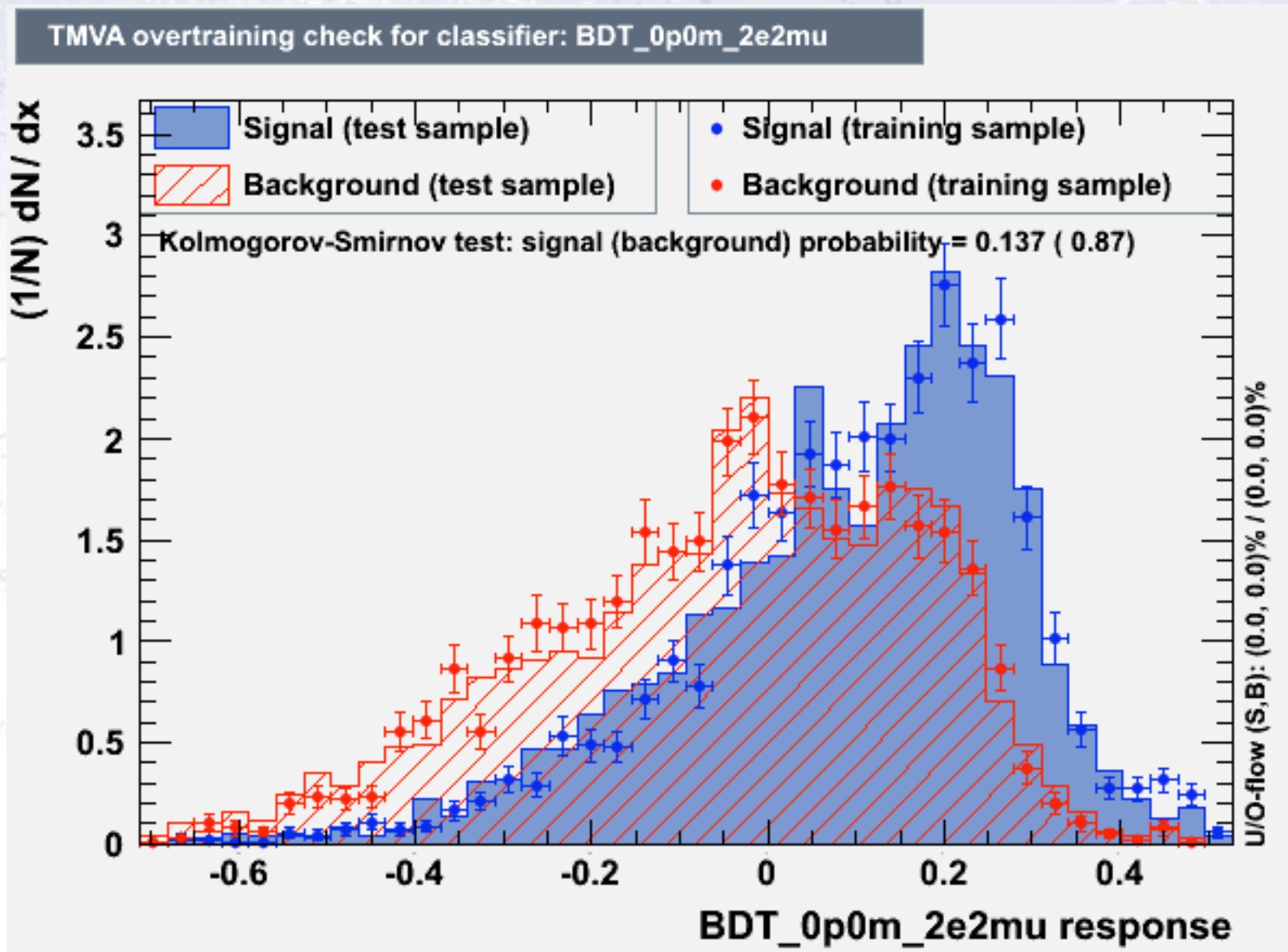
Individual tree





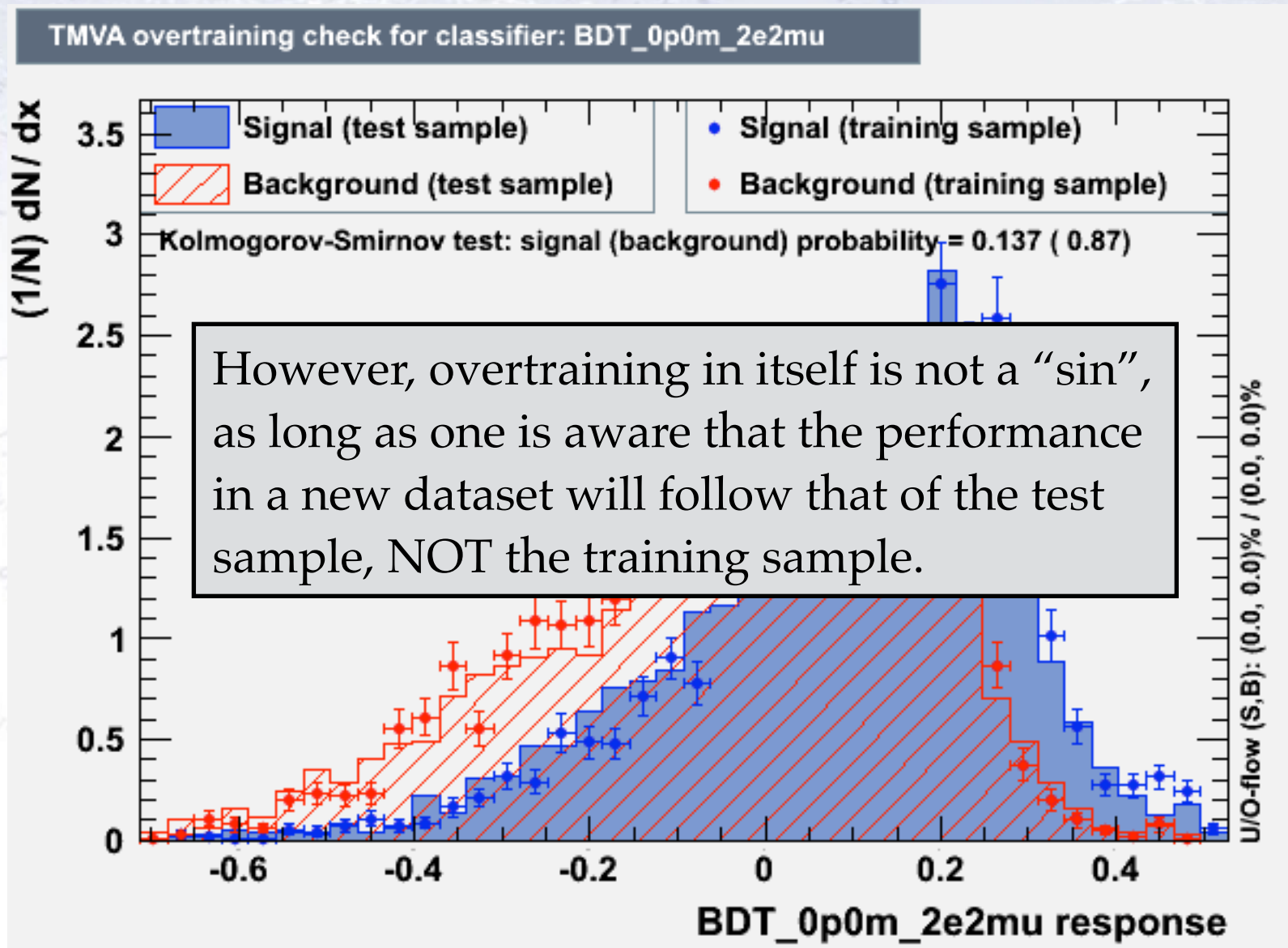
# Test for simple overtraining

In order to test for overtraining, half the sample is used for training, the other for testing:



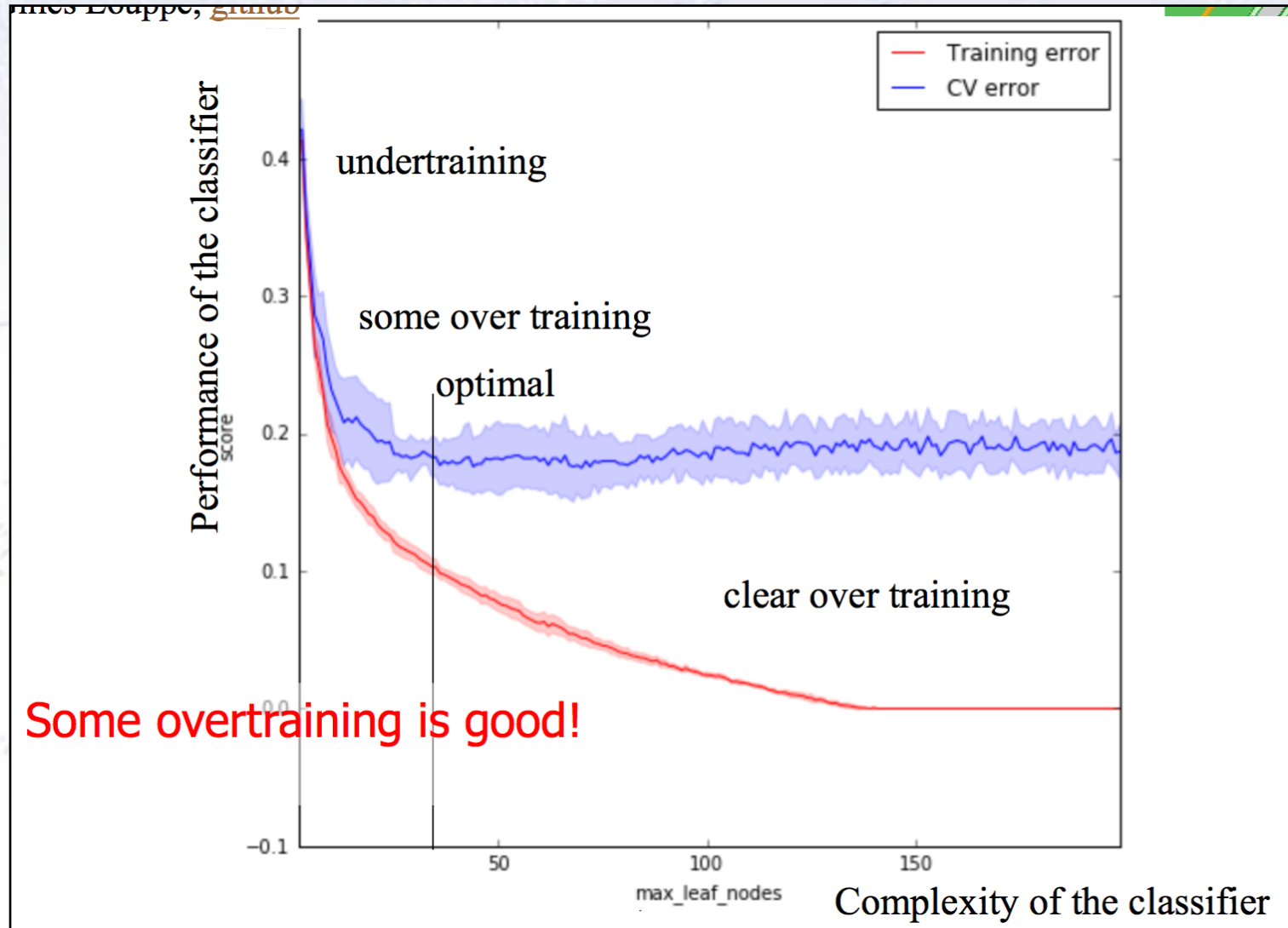
# Test for simple overtraining

In order to test for overtraining, half the sample is used for training, the other for testing:



# Real overtraining

The “real” limit of overtraining, is when the Cross Validation (CV) error starts to grow!



# Method's (dis-)advantages

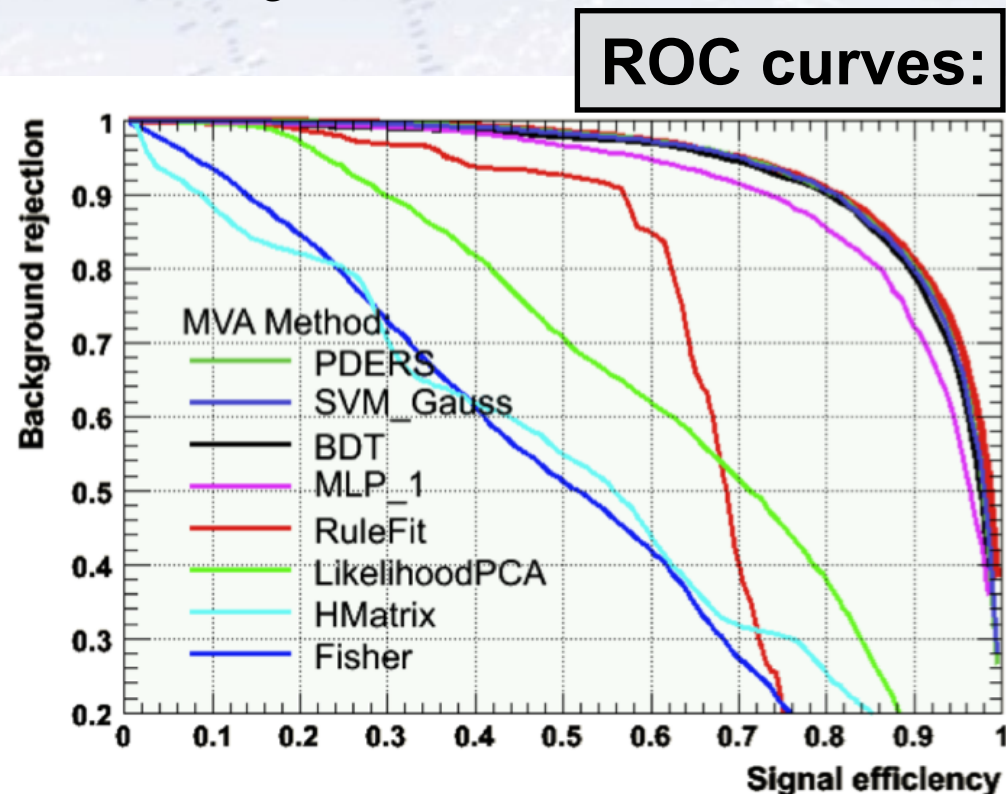
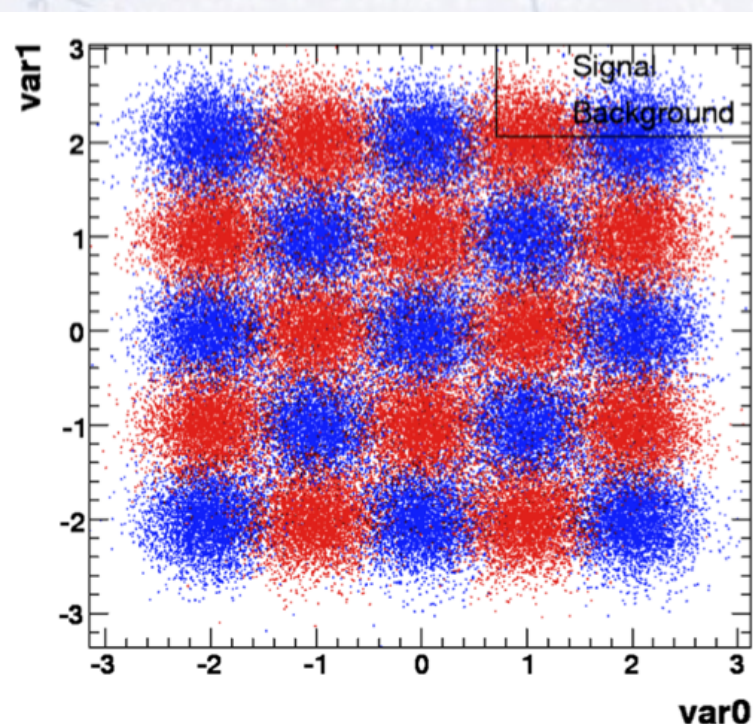
		CLASSIFIERS									
CRITERIA		Cuts	Likeli- hood	PDE- RS	k-NN	H- Matrix	Fisher	ANN	BDT	Rule- Fit	SVM
Performance	No or linear correlations	*	**	*	*	*	**	**	*	**	*
	Nonlinear correlations	o	o	**	**	o	o	**	**	**	**
Speed	Training	o	**	**	**	**	**	*	o	*	o
	Response	**	**	o	*	**	**	**	*	**	*
Robustness	Overtraining	**	*	*	*	**	**	*	o	*	**
	Weak variables	**	*	o	o	**	**	*	**	*	*
Curse of dimensionality		o	**	o	o	**	**	*	*	*	
Transparency		**	**	*	*	**	**	o	o	o	o

**Table 1:** Assessment of classifier properties. The symbols stand for the attributes “good” (\*\*), “fair” (\*) and “bad” (o). “Curse of dimensionality” refers to the “burden” of required increase in training statistics and processing time when adding more input variables. See also comments in text. The FDA classifier is not represented here since its properties depend on the chosen function.



# Example of method comparison

Left figure shows the distribution of signal and background used for test.  
Right figure shows the resulting separation using various MVA methods.



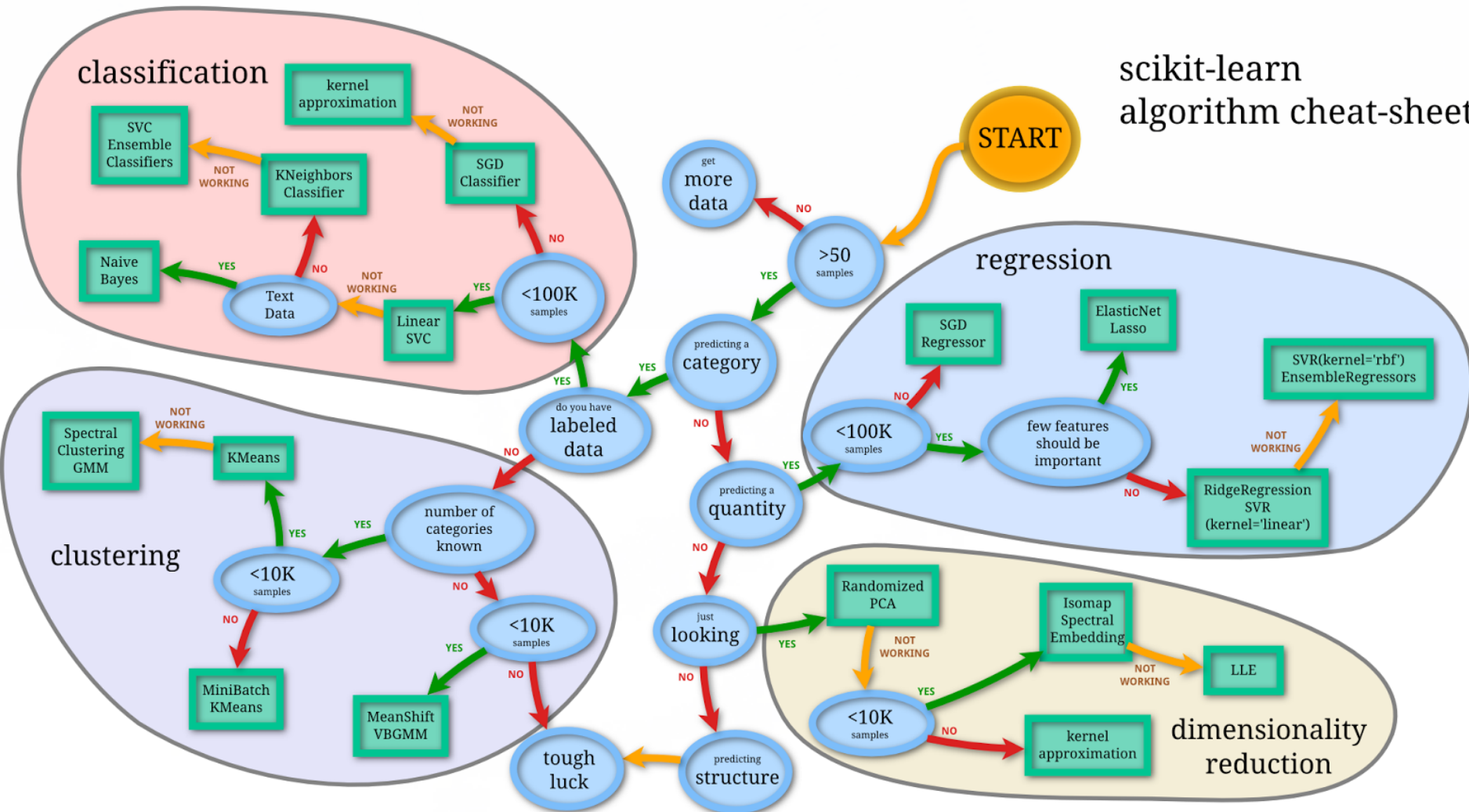
The theoretical limit is known from the Neyman-Pearson lemma using the (known/correct) PDFs in a likelihood.

In all fairness, this is a case that is great for the BDT...

# Which method to use?

There is no good / simple answer to this, though people have tried, e.g.:

scikit-learn  
algorithm cheat-sheet





# Which method to use?

There is no good / simple answer to this, though people have tried, e.g.:

scikit-learn  
algorithm cheat-sheet

