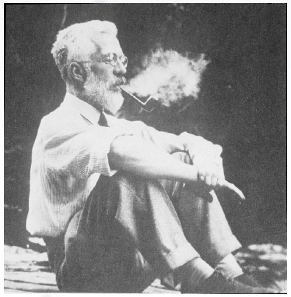
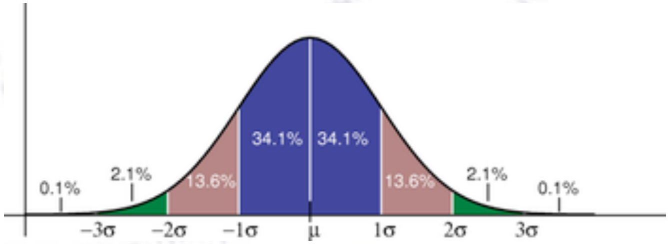
Big Data Analysis

Introduction to MultiVariate Analysis



Troels C. Petersen (NBI)



"Statistics is merely a quantisation of common sense - Machine Learning 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.

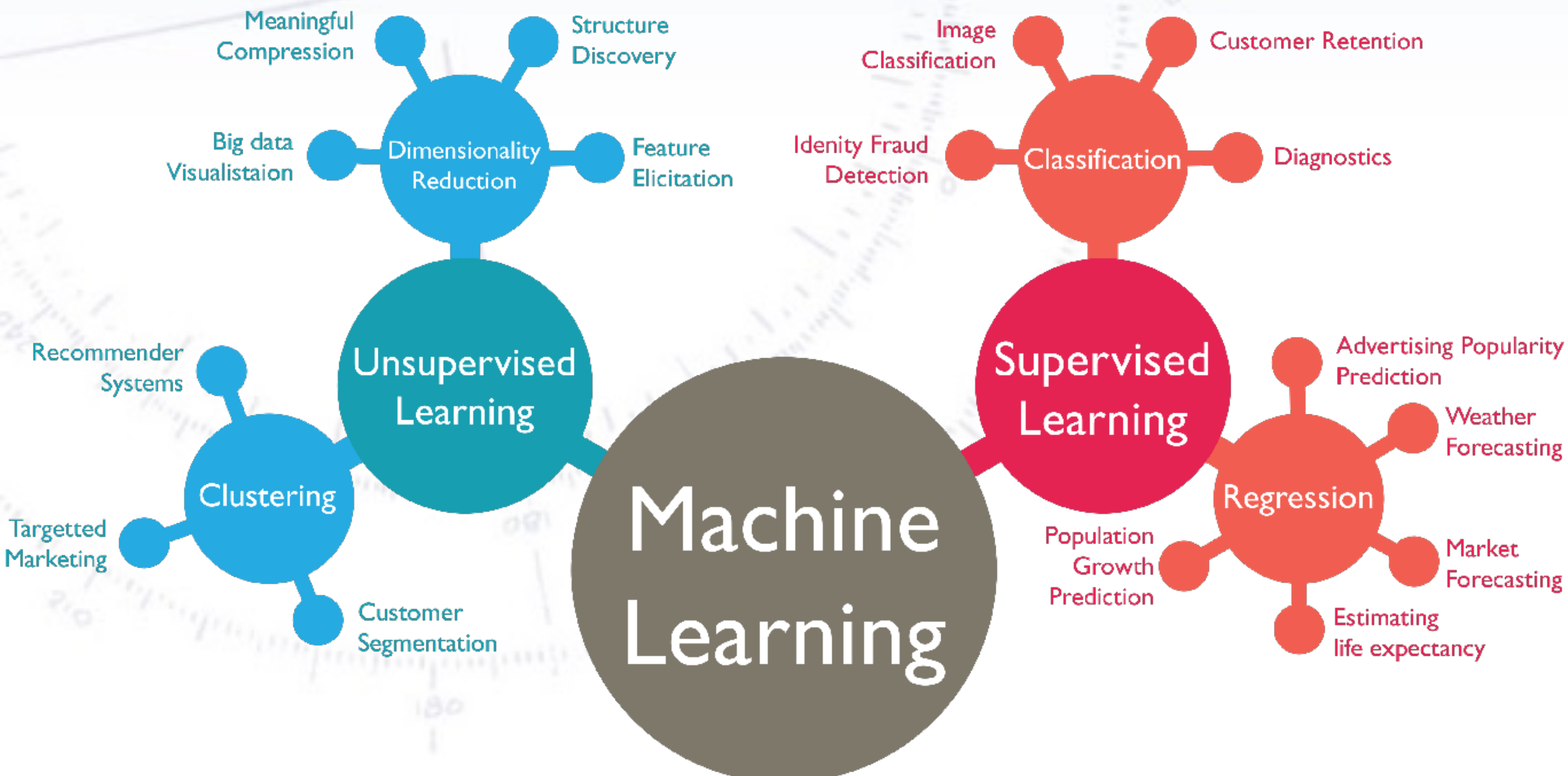
	Low dim.	High dim.
Linear	Humans: ✓ Computers: ✓	Humans: ÷ Computers: ✓
Non-linear	Humans: ✓ Computers: (✓)	Humans: ÷ Computers: (✓)

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!

Classification vs. Regression
Unsupervised learning vs. supervised

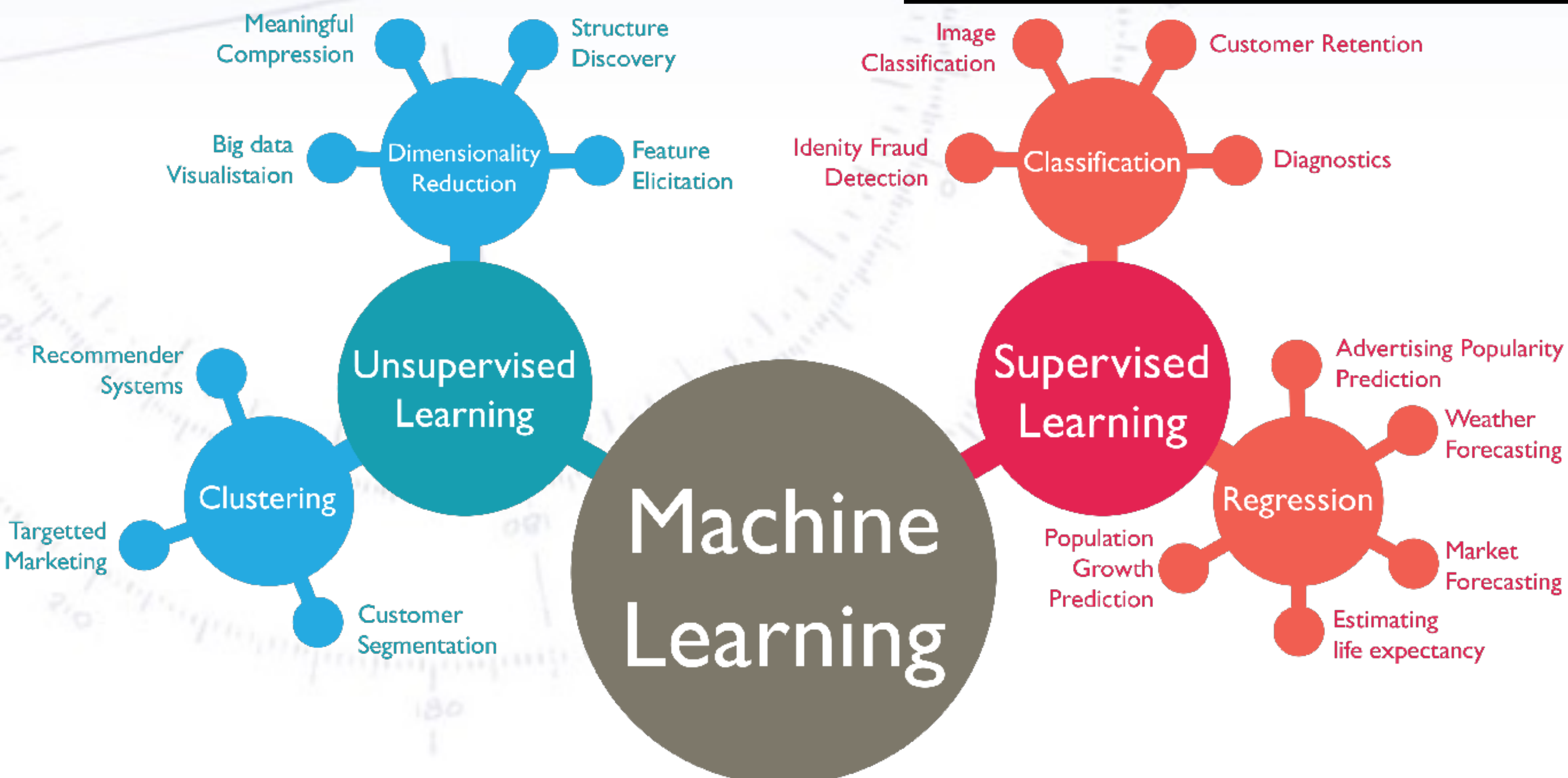
Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)… [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).



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).

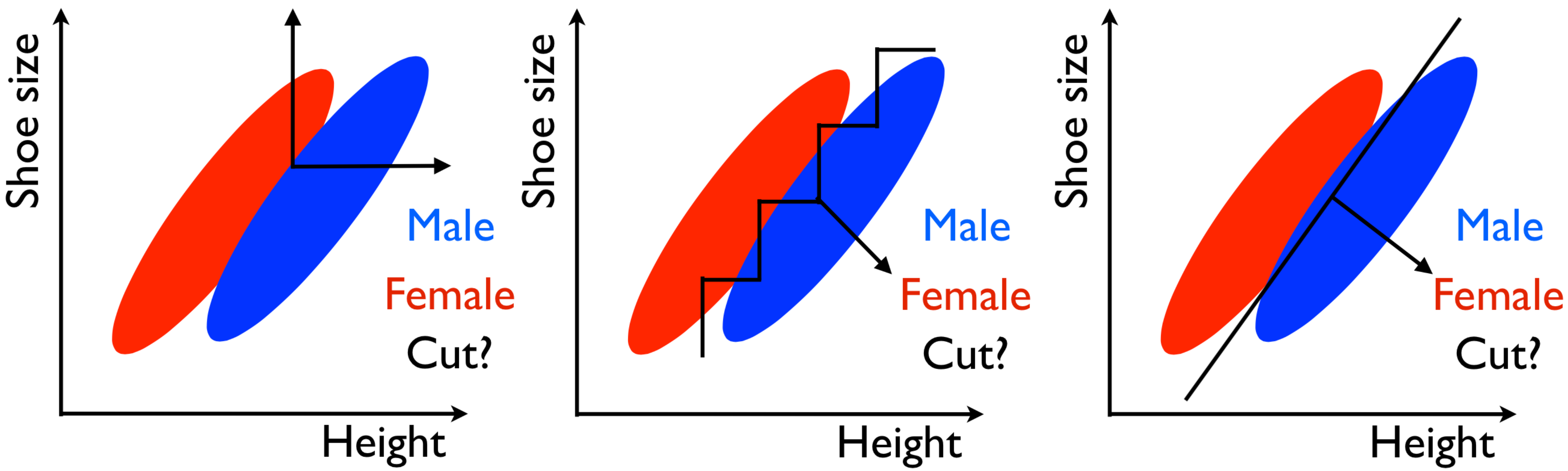
We will be mostly on this side!



Simple Example

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

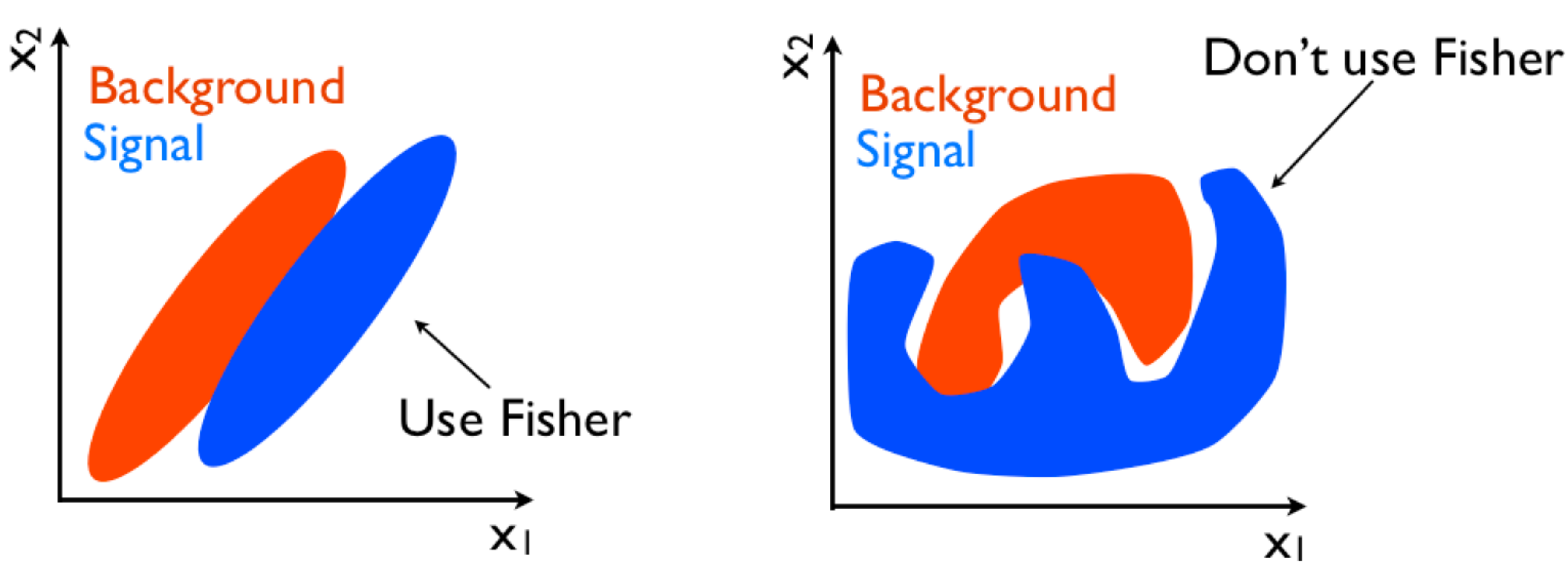
Cut on each var?	Advanced cut?	Combine var?
Poor efficiency!	Clumsy and hard to implement	Smart and promising



The latter approach is the Fisher discriminant!
It has the advantage of being simple and applicable in many dimensions easily!

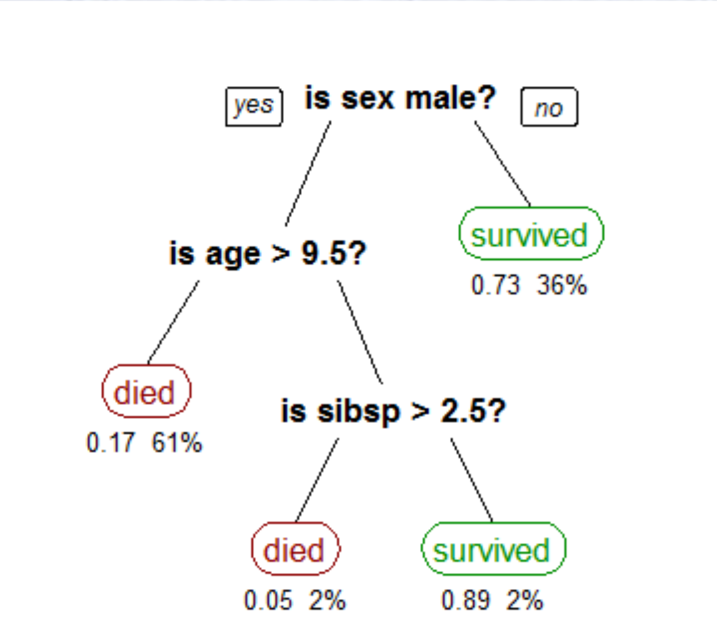
Non-linear MVAs

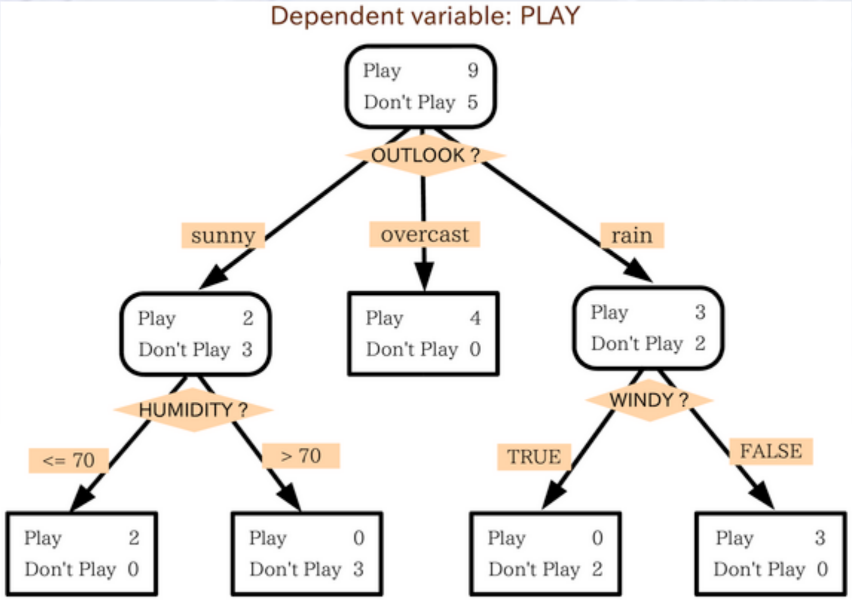
While the Fisher Discriminant uses all separations and **linear correlations**, it does not perform optimally, when there are **non-linear correlations** present:



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...

Decision Trees (DT)



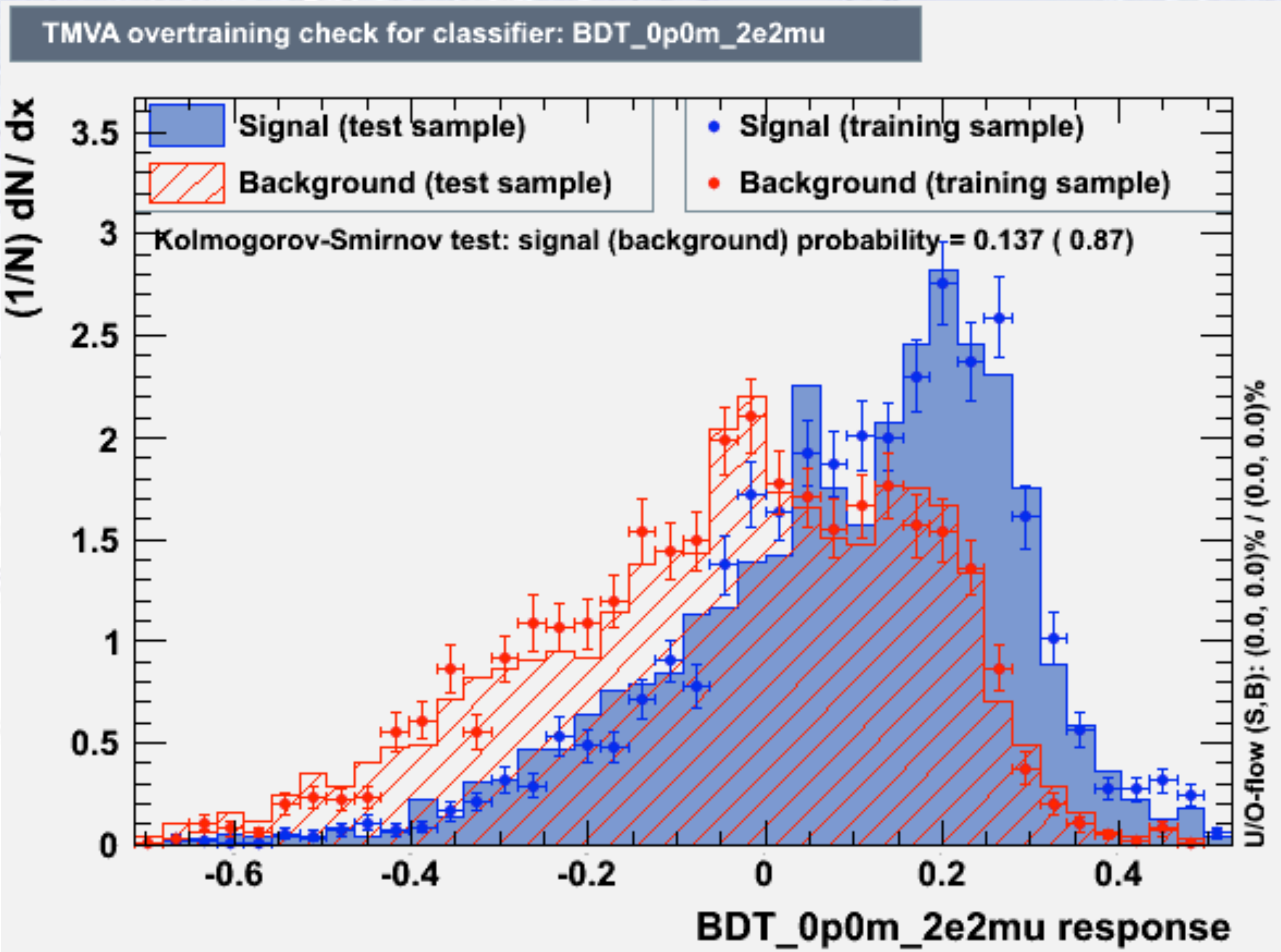


***Decision tree learning** uses a **decision tree** as a **predictive model** which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in **statistics**, **data mining** and **machine learning**.*

[Wikipedia, Introduction to Decision Tree Learning]

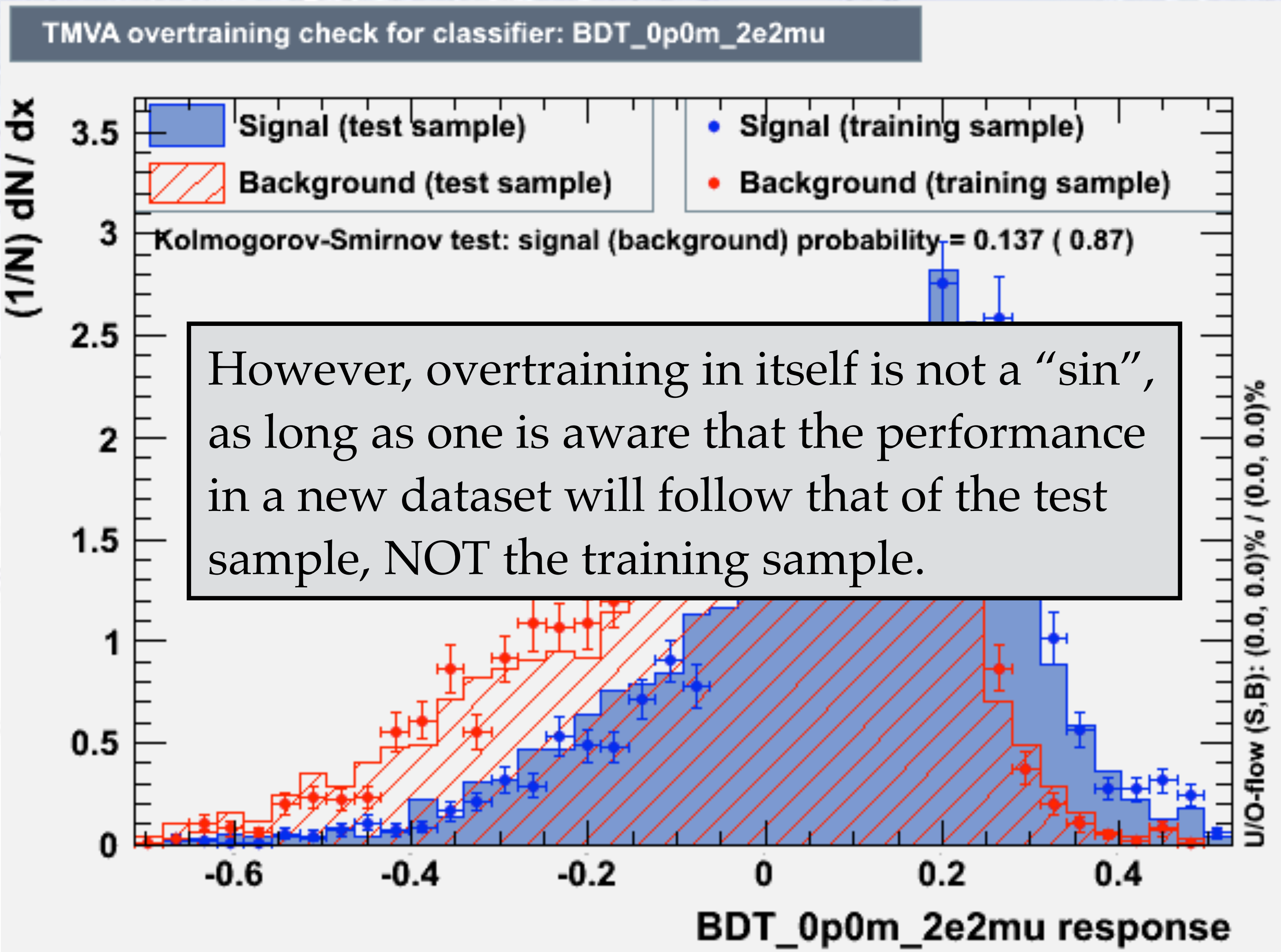
Test for overtraining

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



Test for overtraining

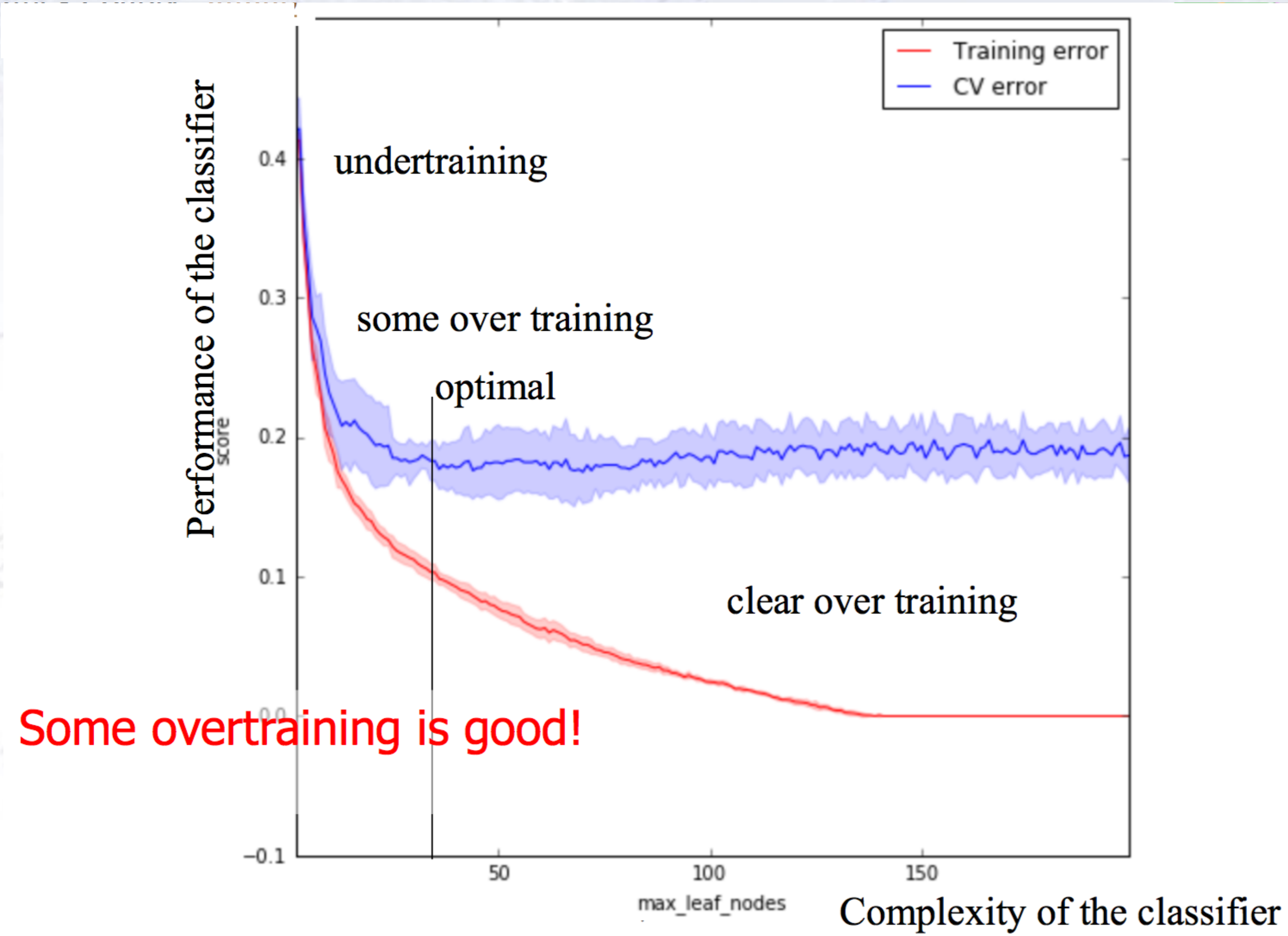
In order to test for overtraining, half the sample is used for training, the other for testing:



However, overtraining in itself is not a "sin", as long as one is aware that the performance in a new dataset will follow that of the test sample, NOT the training sample.

Overtraining…

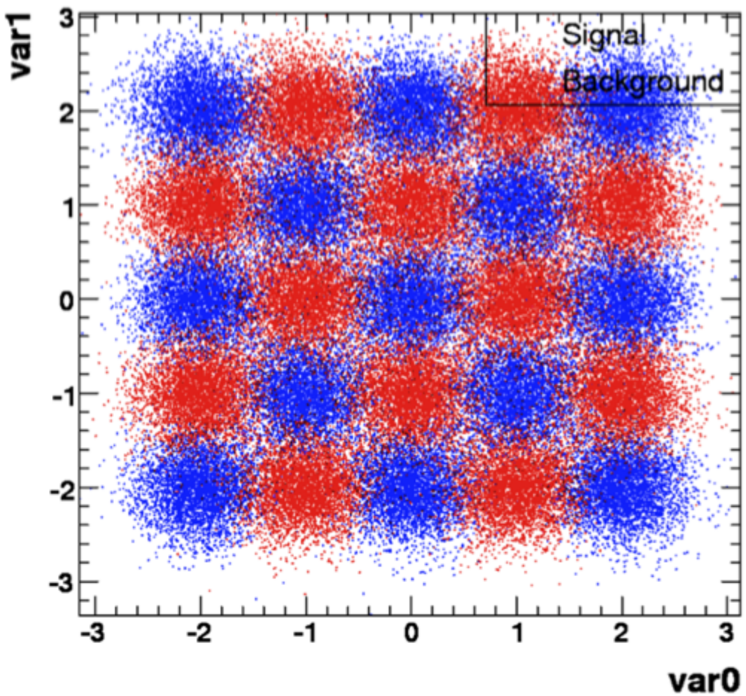
To test for overtraining, try to increase the number of parameters of your ML.
If performance on Cross Validation (CV) sample drops, decrease complexity!

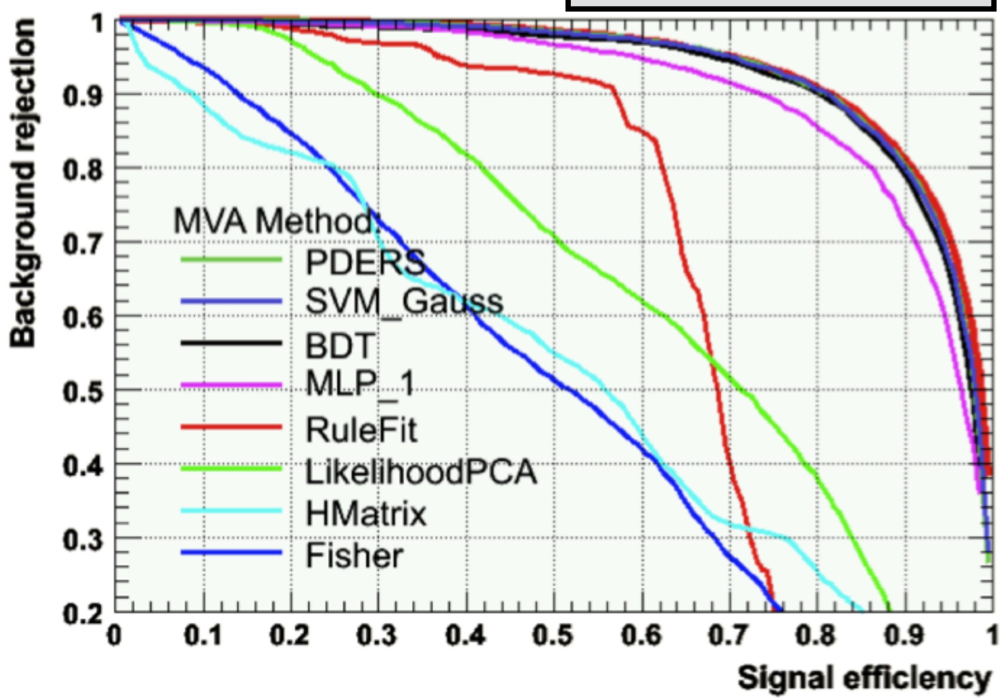


Example of method comparison

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

ROC curves:





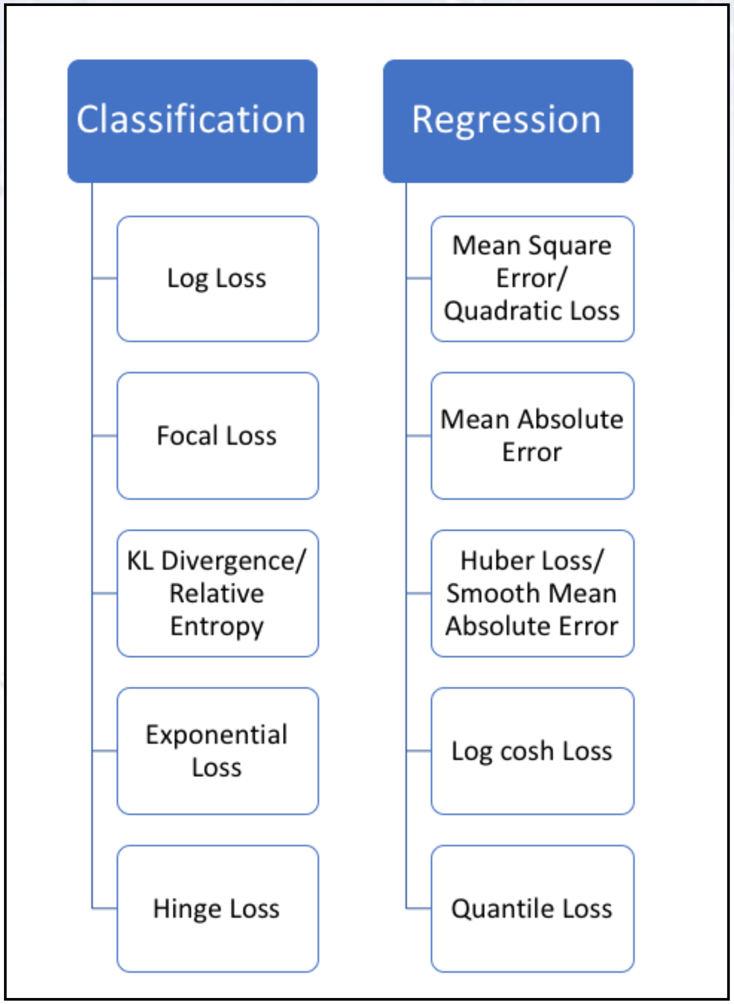
The theoretical limit is known from the Neyman-Pearson lemma using the (known/correct) PDFs in a likelihood.
In all fairness, this is a case that is great for the BDT...

What loss function to use?

The choice of loss function depends on the problem at hand, and in particular what you find important!

In classification:
- Do you care how wrong the wrong are?
- Do you want pure signal or high efficiency?
- Does it matter what type of errors you make?

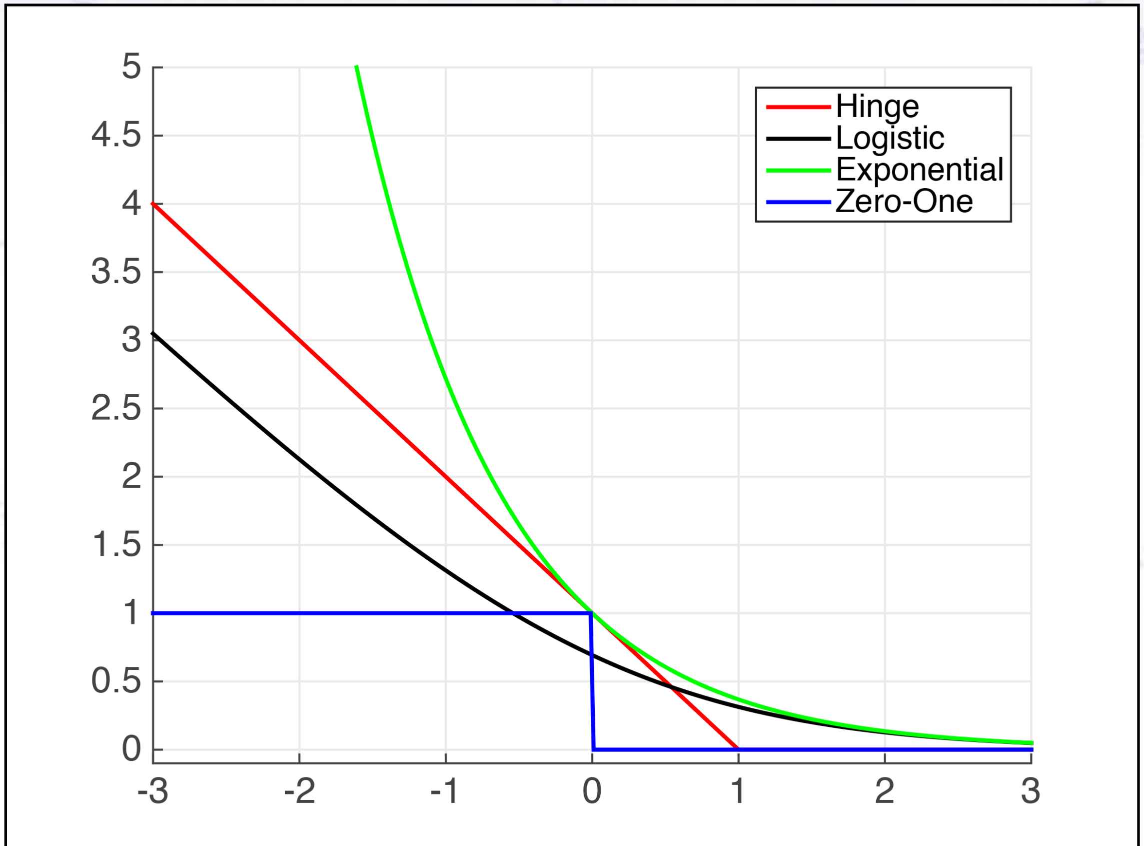
In regression:
- Do you care about outliers?
- Do you care about size of outliers?
- Is core resolution vital?

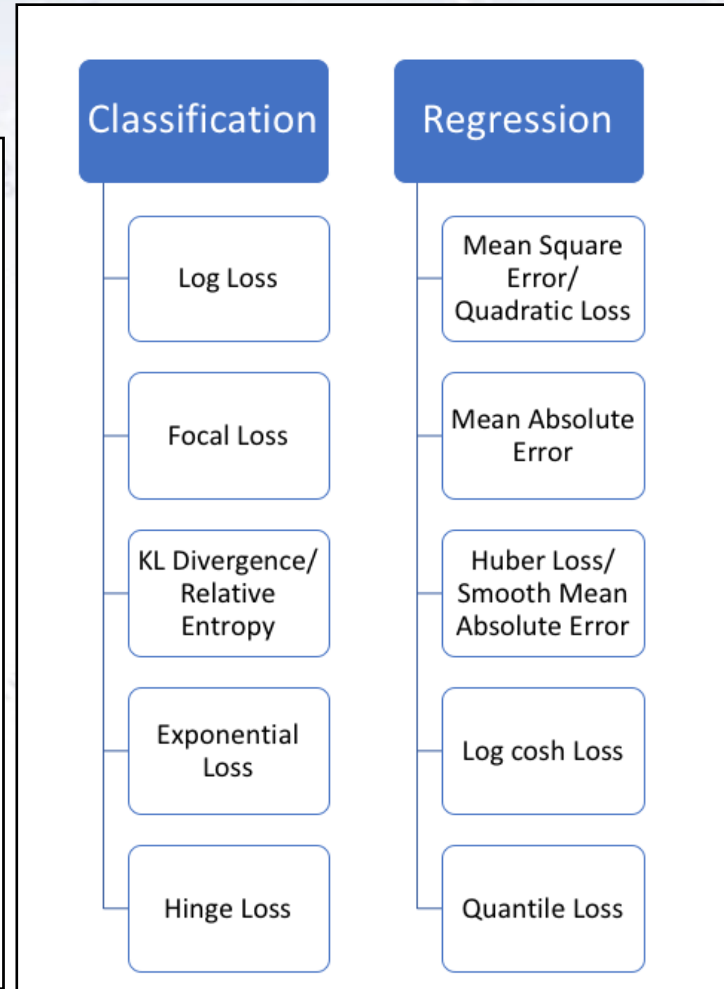


What loss function to use?

The choice of loss function depends on the problem at hand, and in particular what you find important!

Loss functions for classification





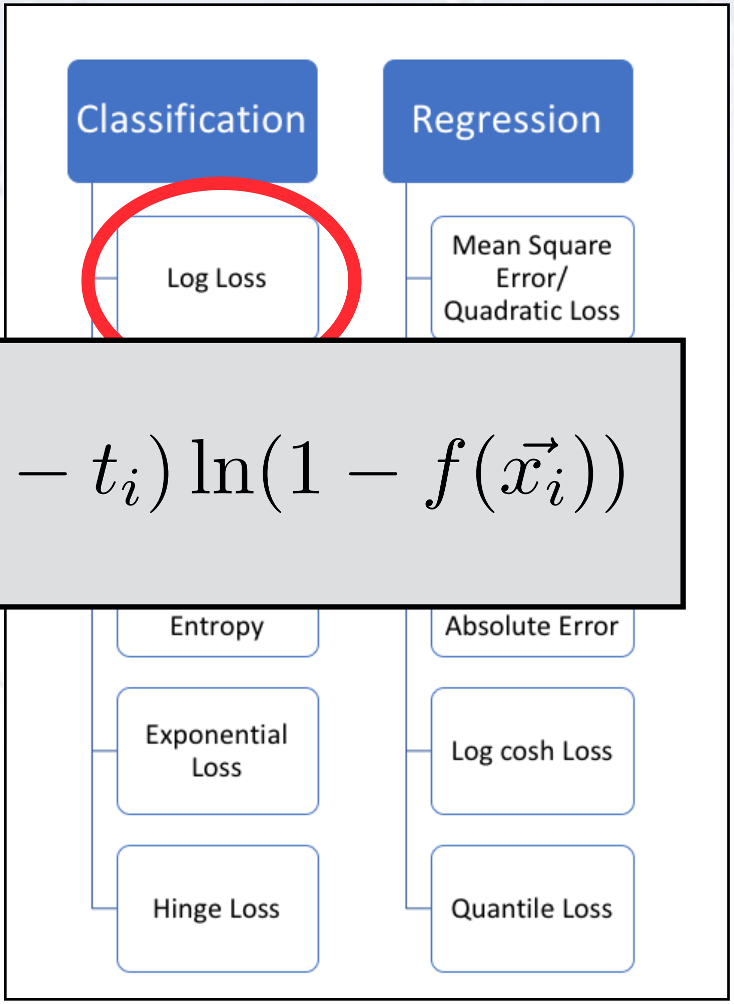
What loss function to use?

The choice of loss function depends on the problem at hand, and in particular what you find important!

Loss functions for classification

A central loss function in classification is the **Cross Entropy**:

$$\text{Loss} = \sum_i -t_i \ln(f(\vec{x_i})) \; - \; (1 - t_i)\ln(1 - f(\vec{x_i}))$$



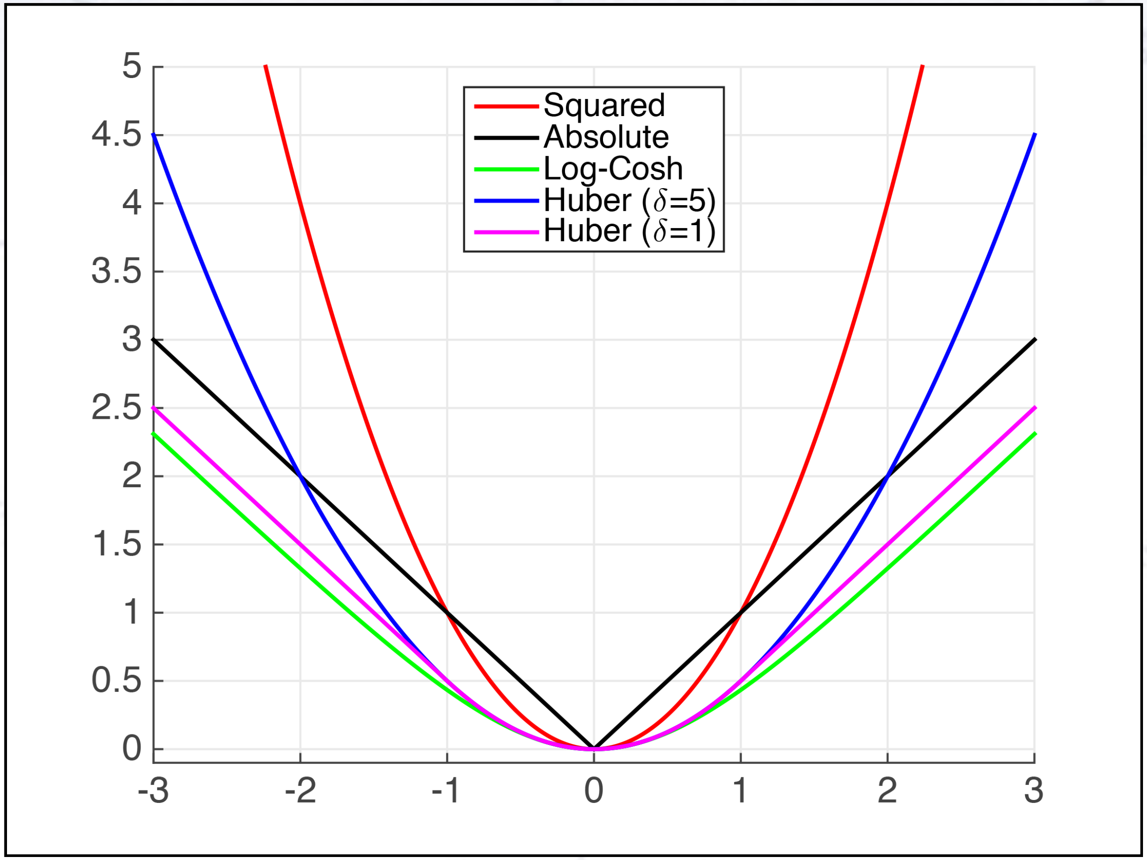
Here, t is the label and f(x) is the classification.

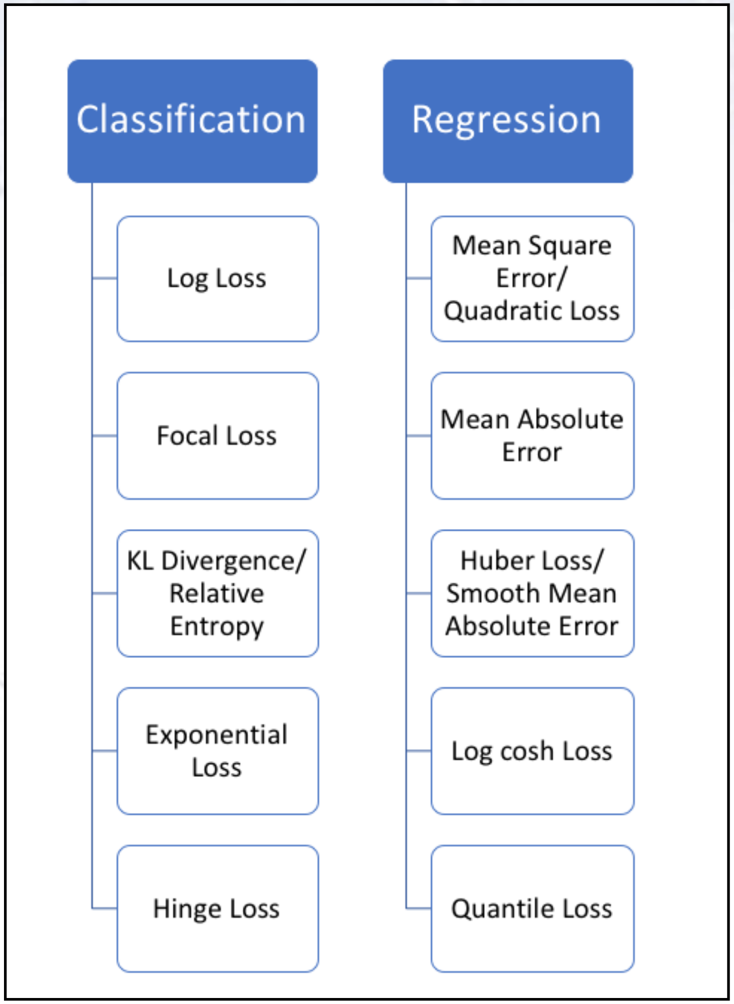
It is based on information theory (Kullback-Leibler divergence), can be minimised using stochastic gradient descent, and plays a central role in deep learning.

What loss function to use?

The choice of loss function depends on the problem at hand, and in particular what you find important!

Loss functions for regression





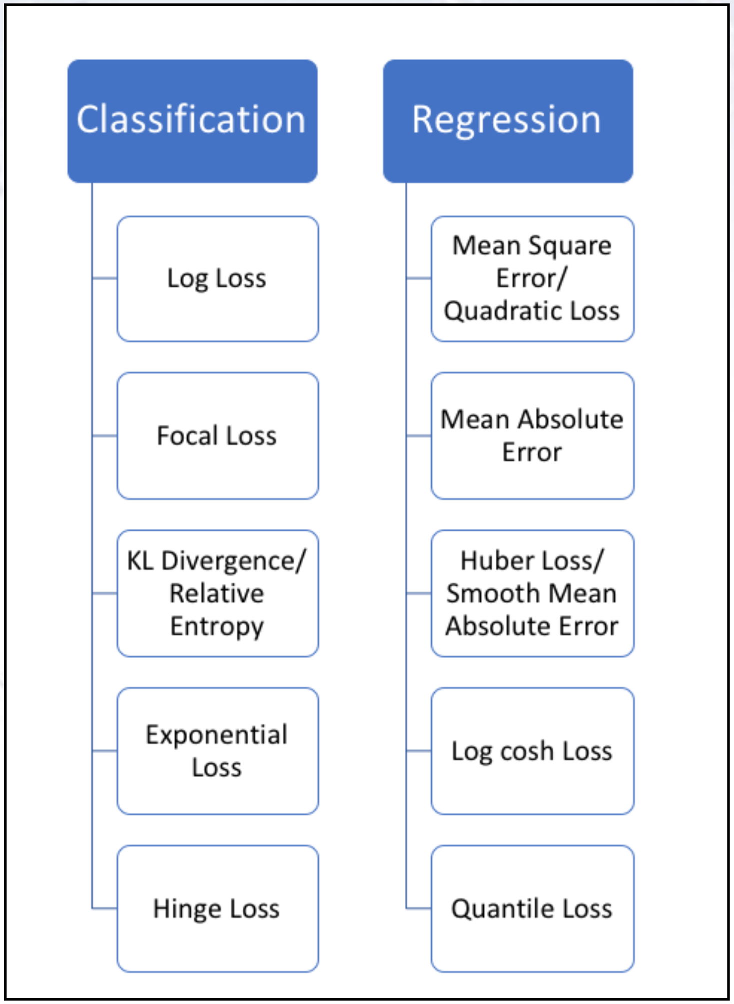
Discussion of regression loss functions

What loss function to use?

The choice of loss function depends on the problem at hand, and in particular what you find important!

In classification:
- Do you care how wrong the wrong are?
- Do you want pure signal or high efficiency?
- Does it matter what type of errors you make?

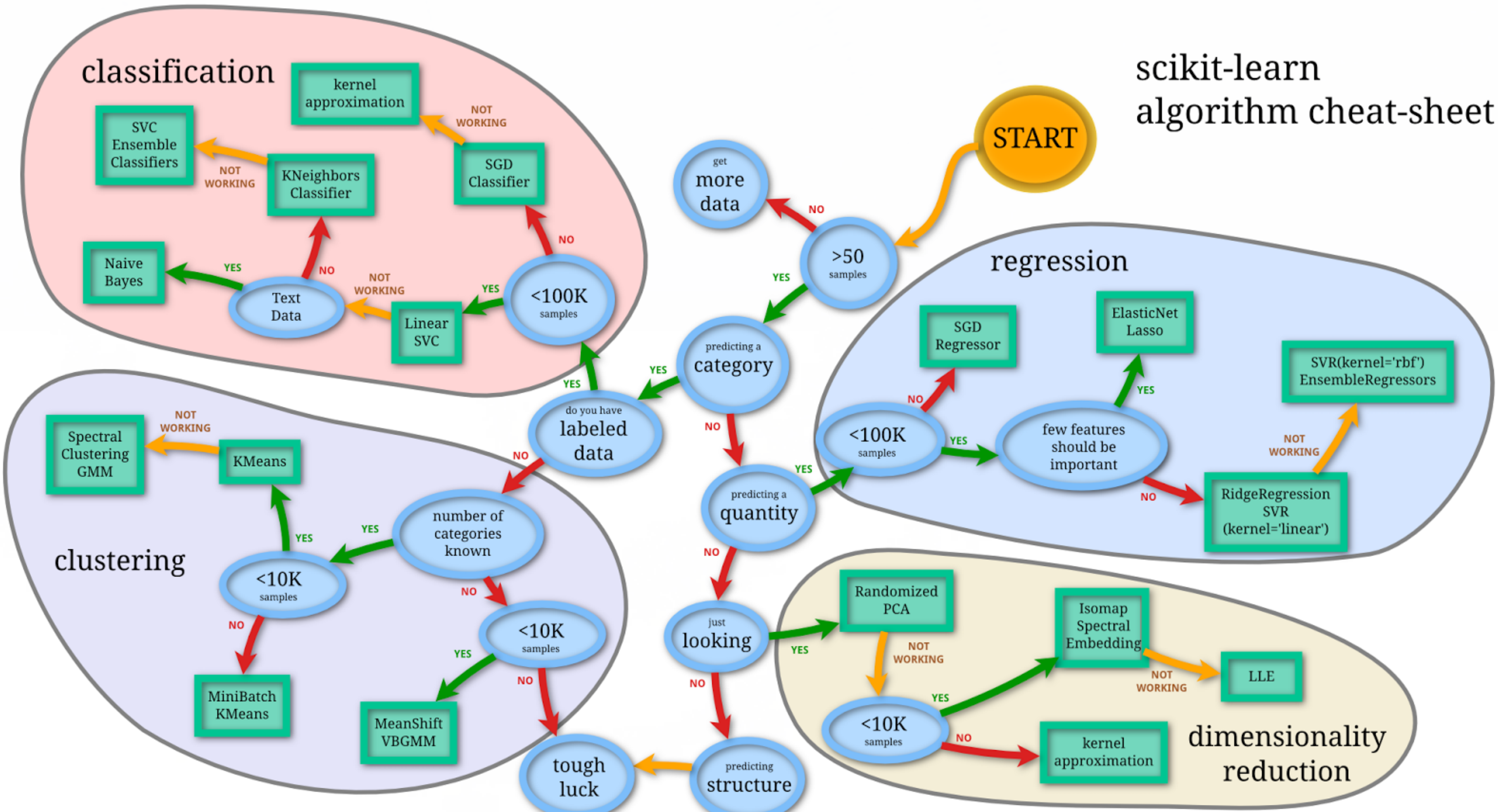
In regression:
- Do you care about outliers?
- Do you care about size of outliers?
- Is core resolution vital?



> Ultimately, the loss function should be tailored to match the wishes of the user. This is however not always that simple, as this might be hard to even know!

Which method to use?

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



Which method to use?

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