Machine Learning Course 2 and tutorial introduction

David Rousseau

IJCLab-Orsay

david.rousseau@in2p3.fr **@dhpmrou**

CHACAL, Johannesburg, Jan 2024

Introduction to classification

Given x, we want y → how to build f ?

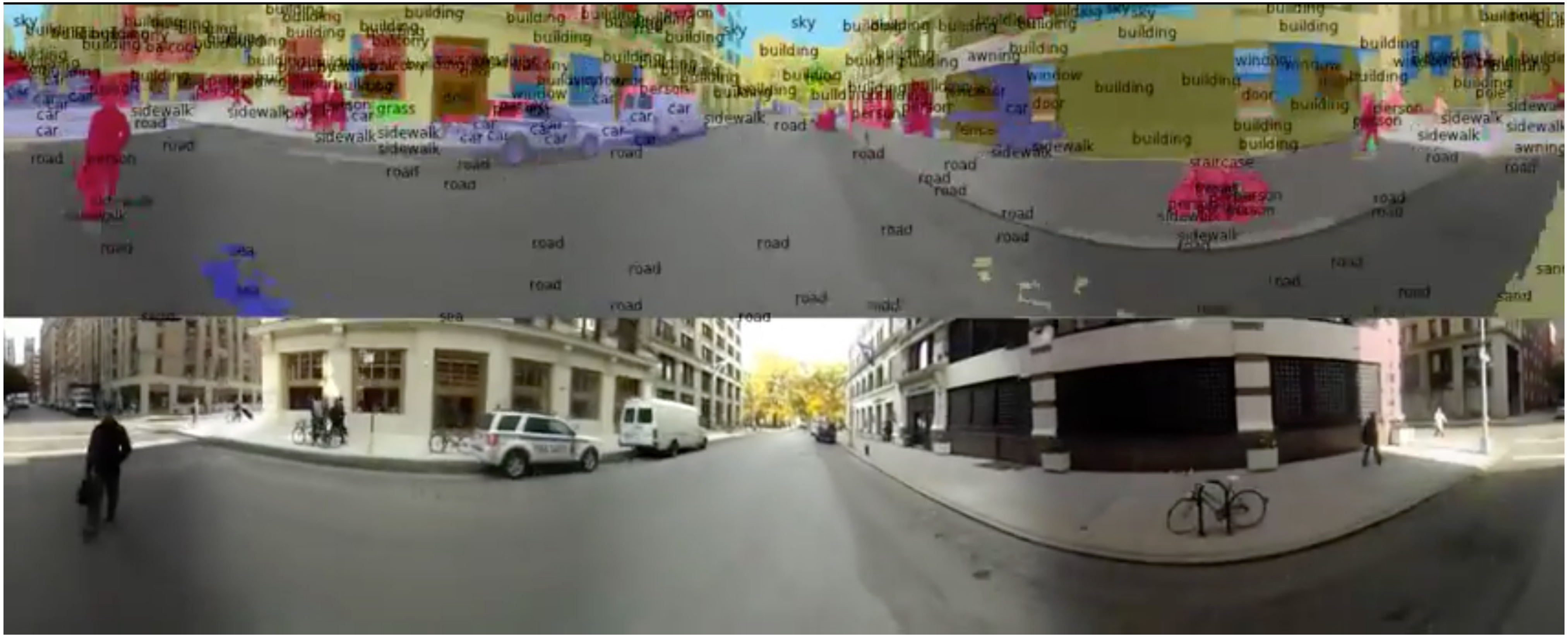
x | f | y

- Written text → text
- Picture → mom or granny?

Classification

- Image → cat or dog?
- « Comment ça va ? » → « Wie geht's ? »
- Speach → text
- Stone positions → next move
- Camera +GPS →steering action
- FB account details → targetted ads

Classification is everywhere

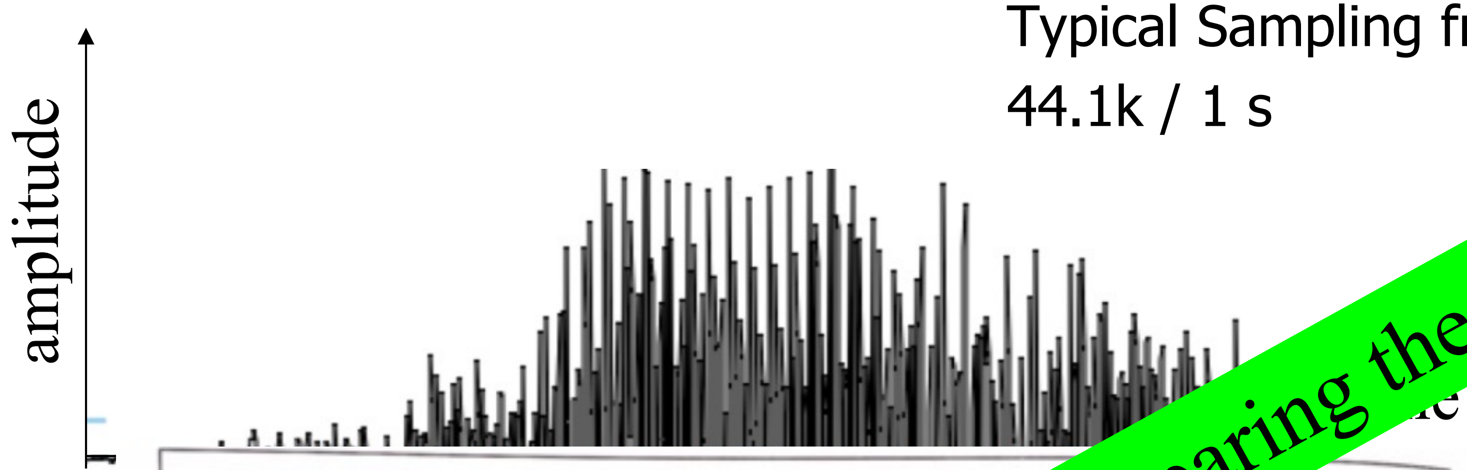


Inputs

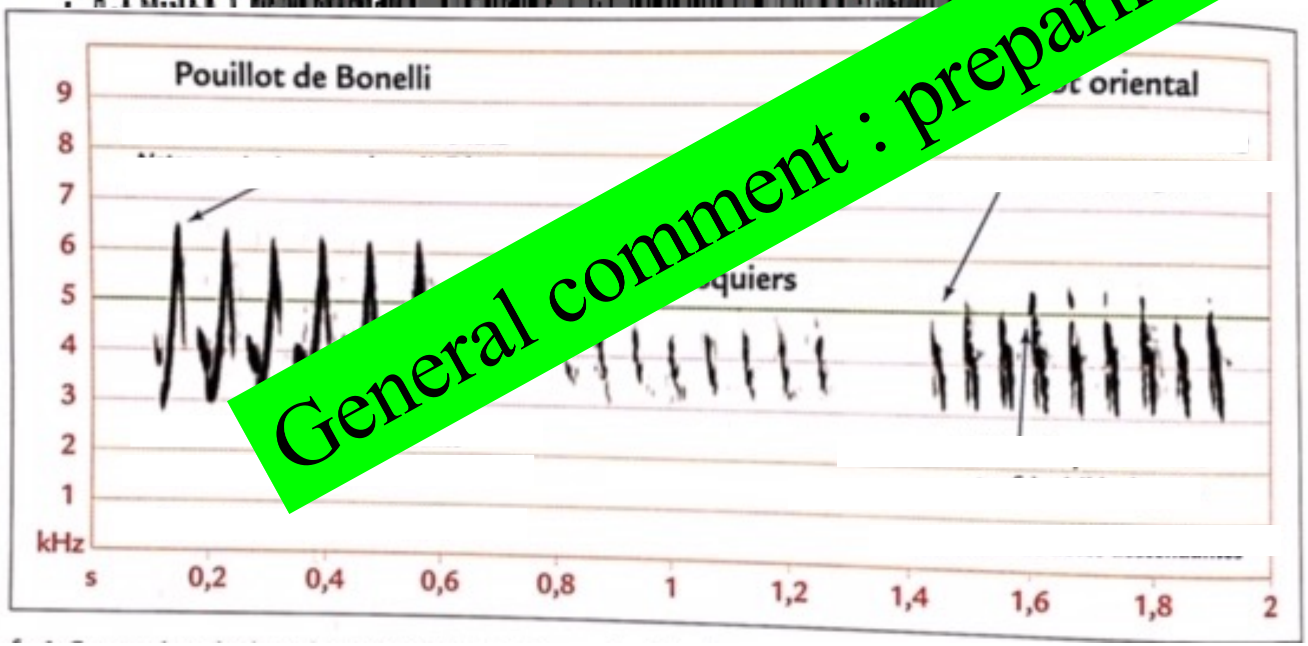
Inputs

Typical Sampling frequency 44.1kHz

44.1k / 1 s



General comment : preparing the data



➔images

Time-frequency diagram

Stanislas Wroza

Tabular Datasets

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa

```
df = pd.read_csv('assets/train.csv')
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Tabular datasets (mostly) in this course

	dijet_invmass	dijet_deltaeta	jet_pt_0	jet_pt_1	eta_zepp_ZZ	min_dR_jZ	pt4ljj_unconstrained	le
39	502.338409	3.694098	140.582153	43.095196	1.529987	1.799823	51.743145	
76	166.194427	0.426846	171.107452	81.588737	0.663560	1.020612	152.570358	
97	269.543396	2.568801	81.123795	64.938507	0.404464	0.050431	50.000000	
107	130.786301	0.119691	171.627014	31.095165	1.329497	0.539539	50.000000	
129	139.976868	2.145803	51.312862	37.323059	3.293238	0.423458	50.000000	

Not tabular datasets

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

Charles Dickens, *A tale of two cities*

- Natural Language is not tabular
- Also features like : « Age of the children »
  - [],[3], [3,7,18]
- Closer to physics : « Energy of the jets in this proton collision »:
  - [],[120.5],[509.2,439.1,123.6,13.3]
- Special techniques to deal with these, see ChatGPT (not in this course)

Output label

- Two classes, usually :
  - y=0 ➔ background
  - y=1 ➔ signal
- N classes :
  - Nearly never used: :
    - y=0➔cat
    - y=1➔dog
    - y=2➔rabbit
  - Rather use « one-hot vector »
    - y=[1,0,0]➔cat
    - y=[0,1,0]➔dog
    - y=[0,0,1]➔rabbit

Classification performance

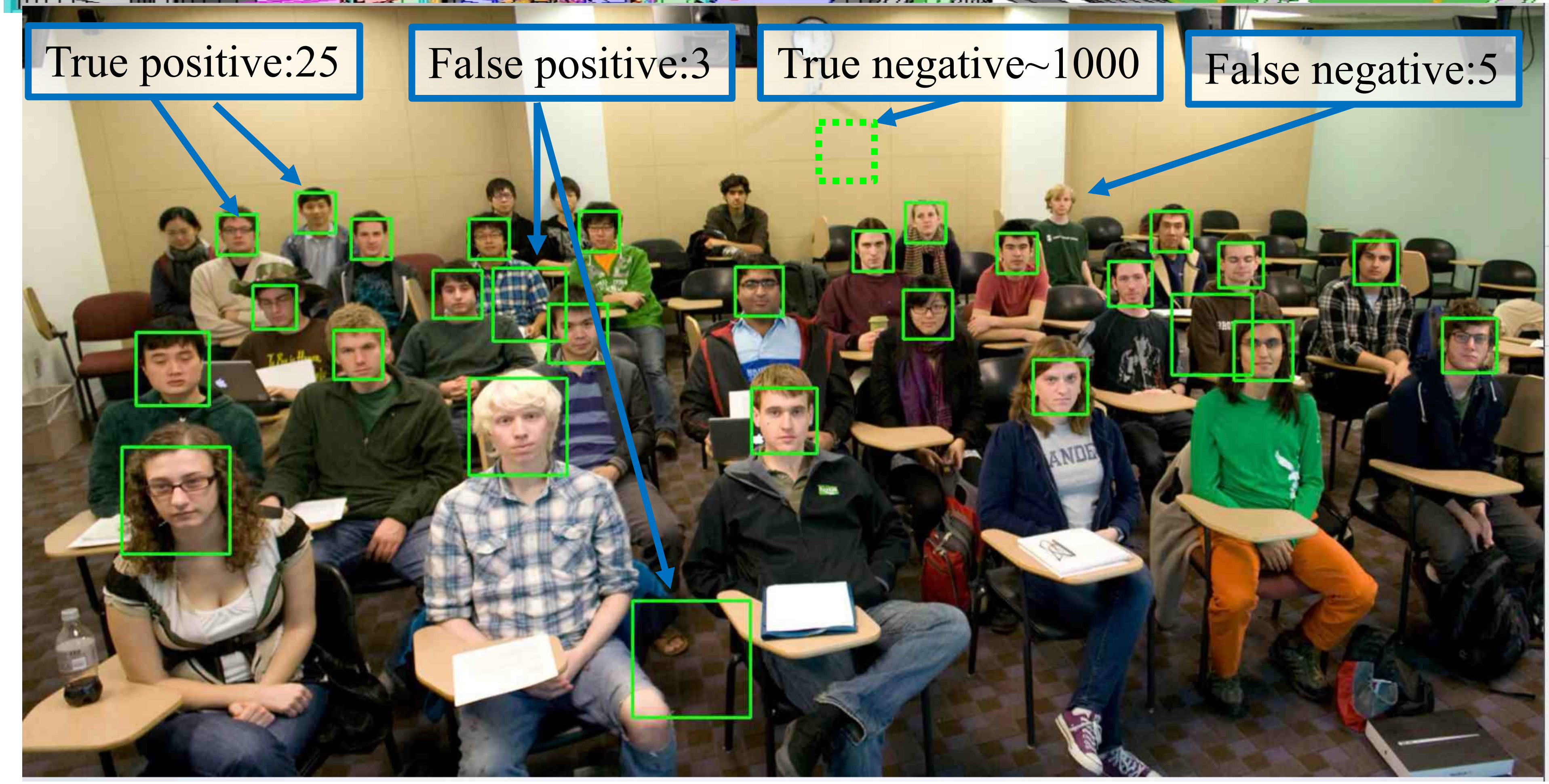
	Signal	Background
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

purity

efficiency

- Total Signal : TP+FN
- Total Background : FP+TN
- Performance numbers
  - (phys) Efficiency==(ML) Recall= TP / (TP+FN)
  - (phys) Purity==(ML) Precision = TP / (TP+FP)

Real-time face detection



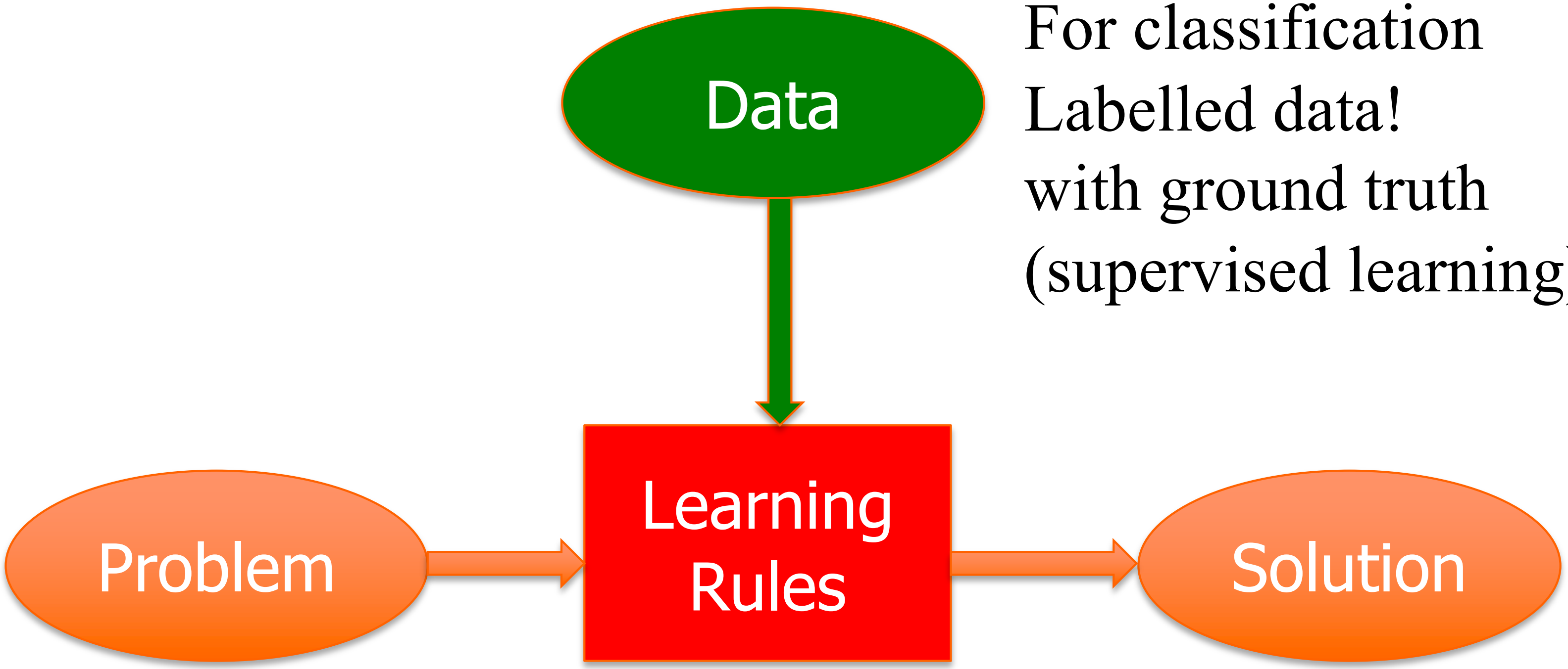
Efficiency==(ML) Recall=83%=25/(25+5)

Purity==(ML) Precision=89%=25/(25+3)

Training dataset

Machine Learning

For classification
Labelled data!
with ground truth
(supervised learning)



Data label example

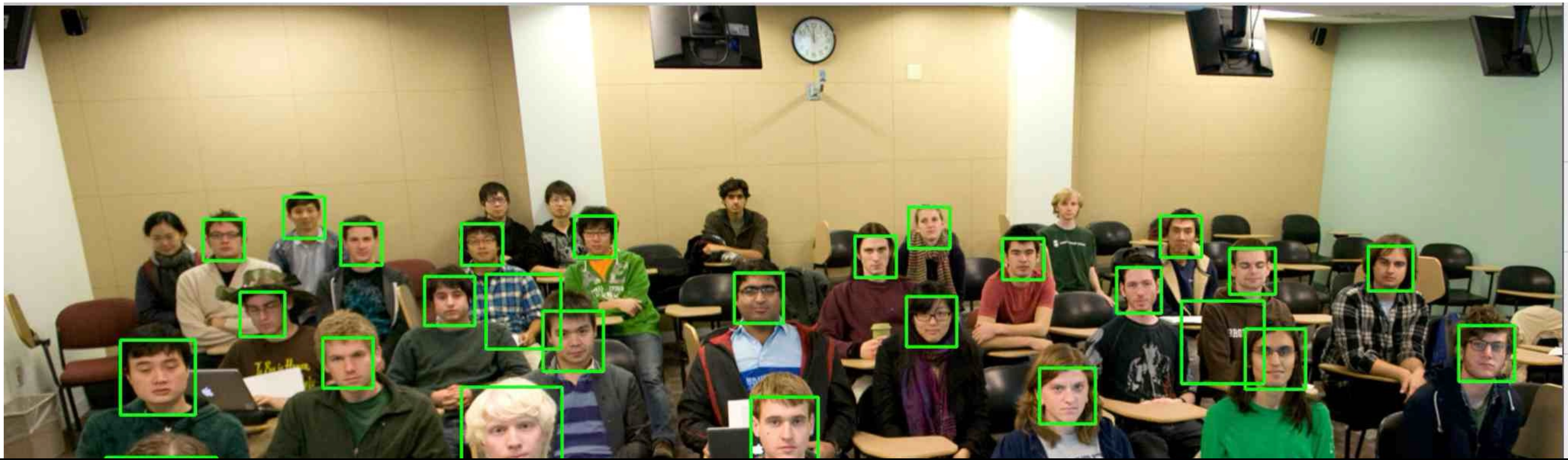
```
df = pd.read_csv('assets/train.csv')
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Data label example

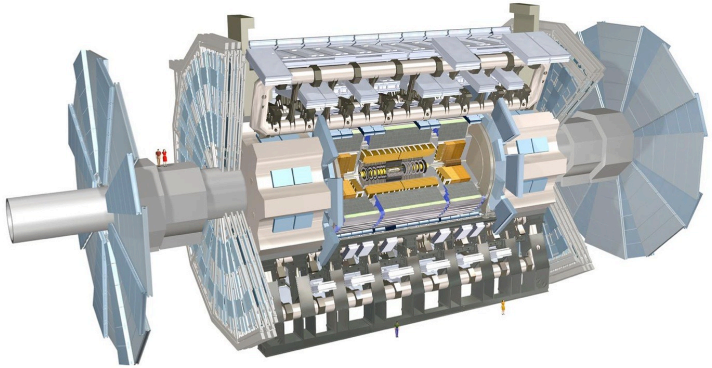
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Data label example

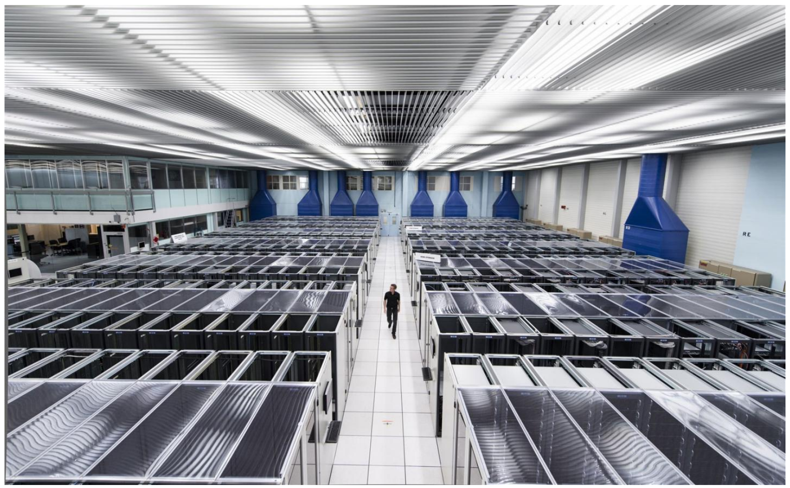


➔most Computer Vision tasks need human labelling!

Data label example



Data



Simulator

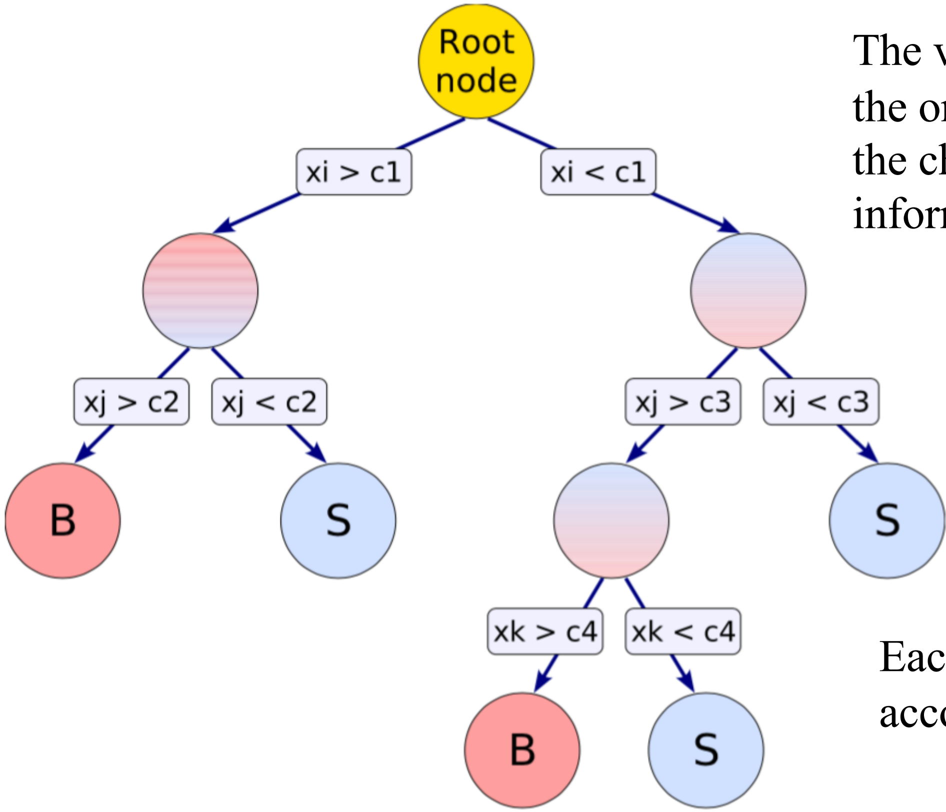
Provide pseudo-data with labels

In a nutshell

- "Classifier"
  - A model F(x)=y : with y=0 or 1 (e.g. 0=Background 1=Signal)
    - In practice : y ranking variable, the larger the more signal-like
  - If size (y)>1: (not in this course)
    - Multiclass : Σ y=1 : Cat or Dog or Elephant
- "Classes"==label : the different categories into which we want to classify. Two categories cases : (A,B), (Signal, Background), (sick, healthy),…
- "Features" == Variables (x)
  - Continuous
  - Discrete
- Classification performance, True/False Positive/Negative
  - Total Signal : TP+FN
  - Total Background : FP+TN
  - (phys) Efficiency==(ML) Recall= TP / (TP+FN)
  - (phys) Purity==(ML) Precision = TP / (TP+FP)
- Training dataset with ground truth : the « true » label==class

Decision Trees

Boosted Decision Tree

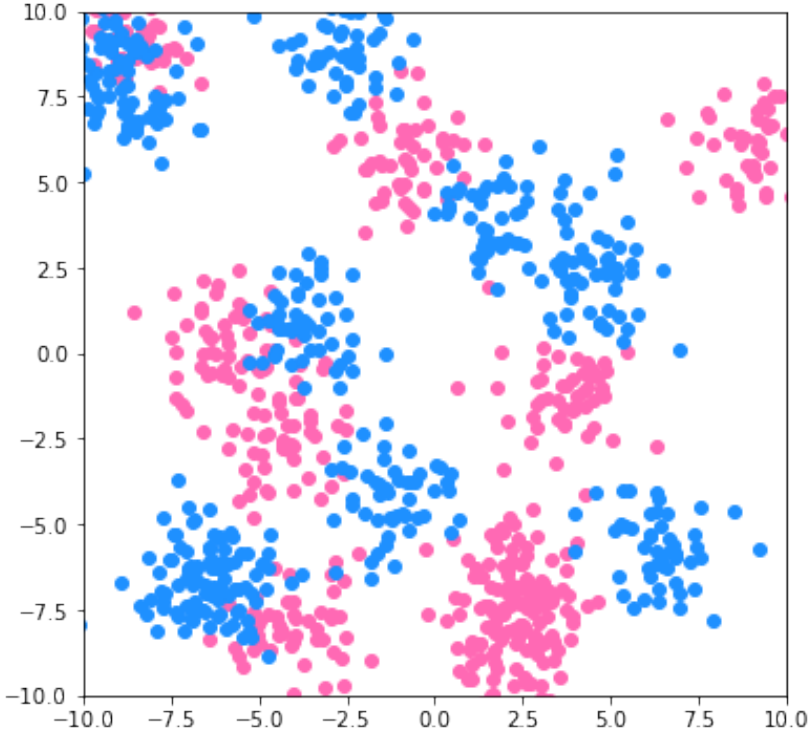


The variable used at each node is the one optimising a measure in the child node (min entropy, information gain…)

Each node given a score according to its purity

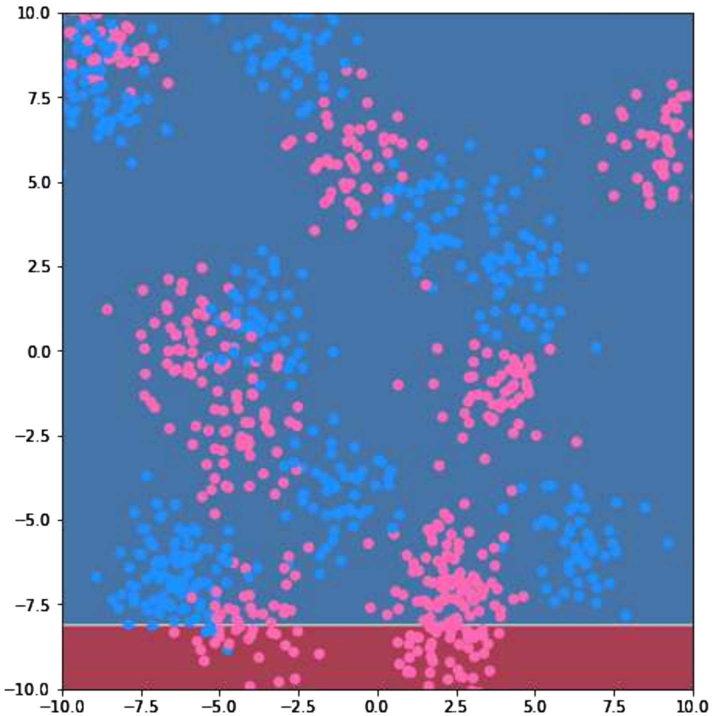
- Single tree (CART) <1980
- AdaBoost 1997 : rerun increasing the weight of misclassified entries ➔Boosted Decision Trees (Gradient BDT XGBoost, random forest…)

Trees at work



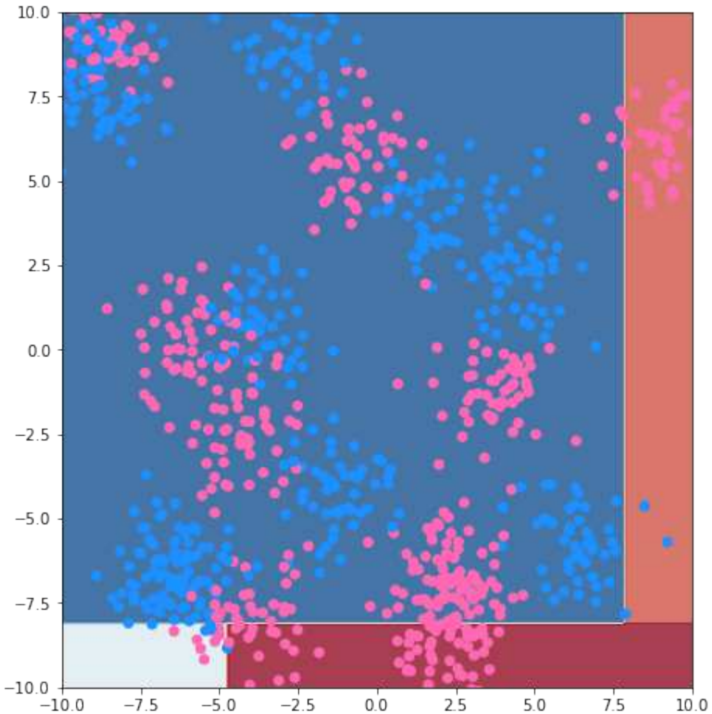
Trees at work

Decision tree, depth=1



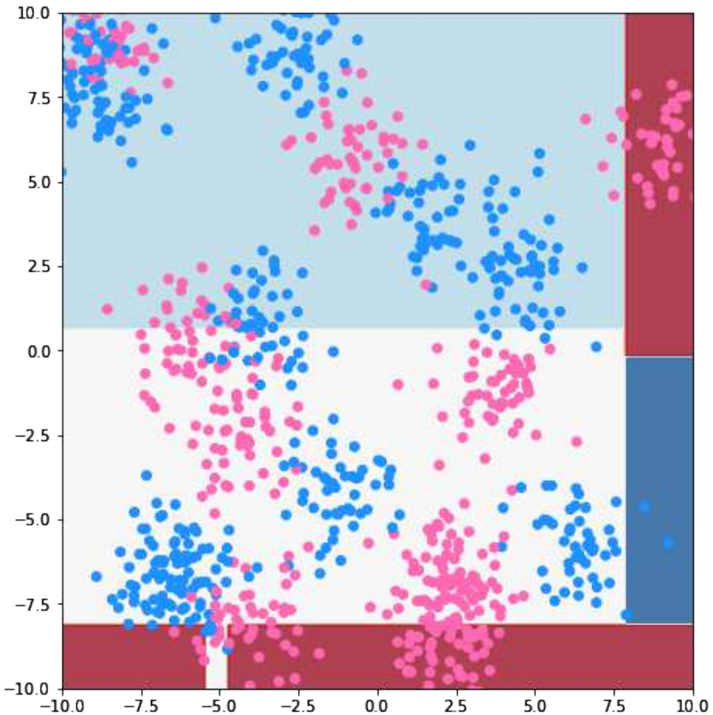
Trees at work

Decision tree, depth=2



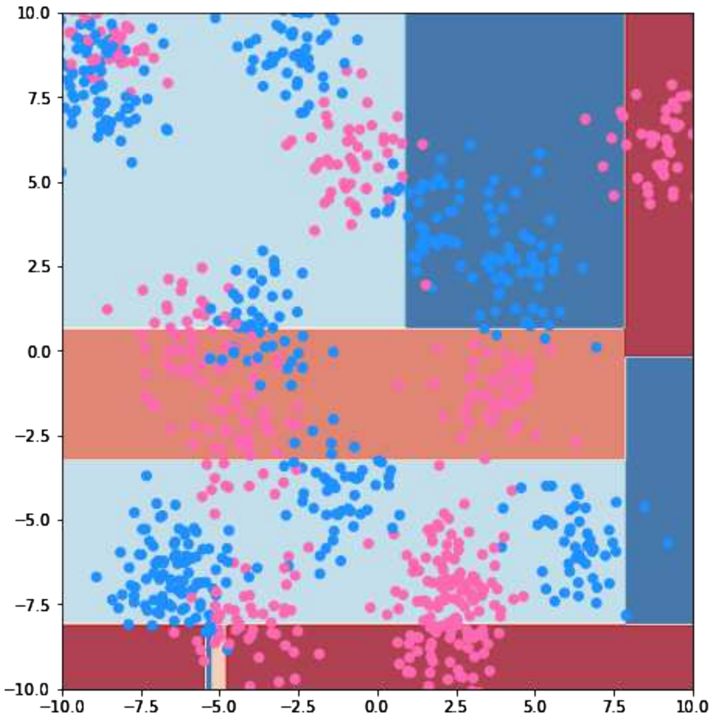
Trees at work

Decision tree, depth=3



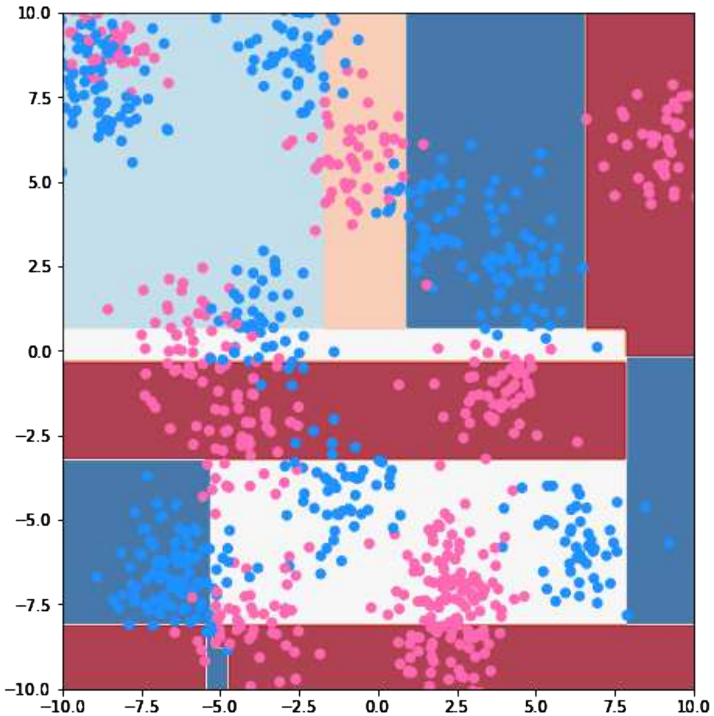
Trees at work

Decision tree, depth=4



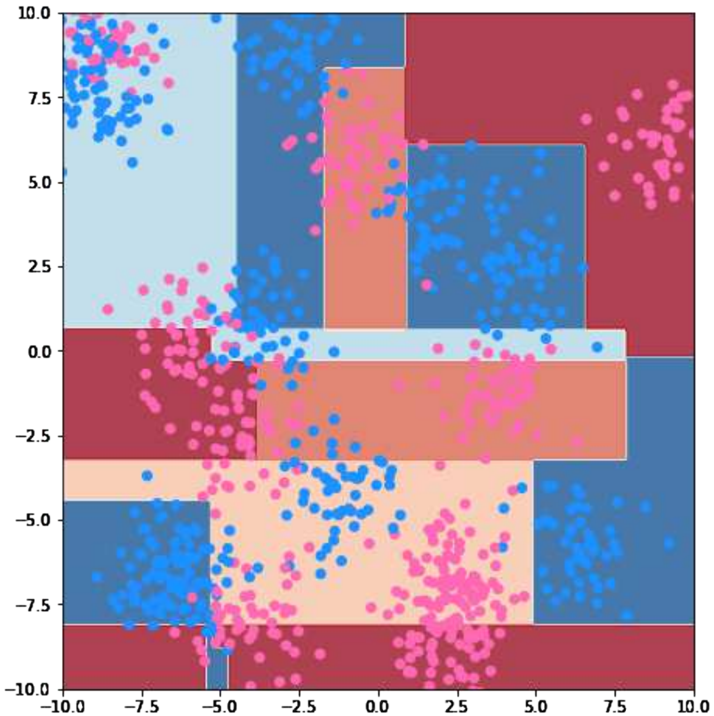
Trees at work

Decision tree, depth=5



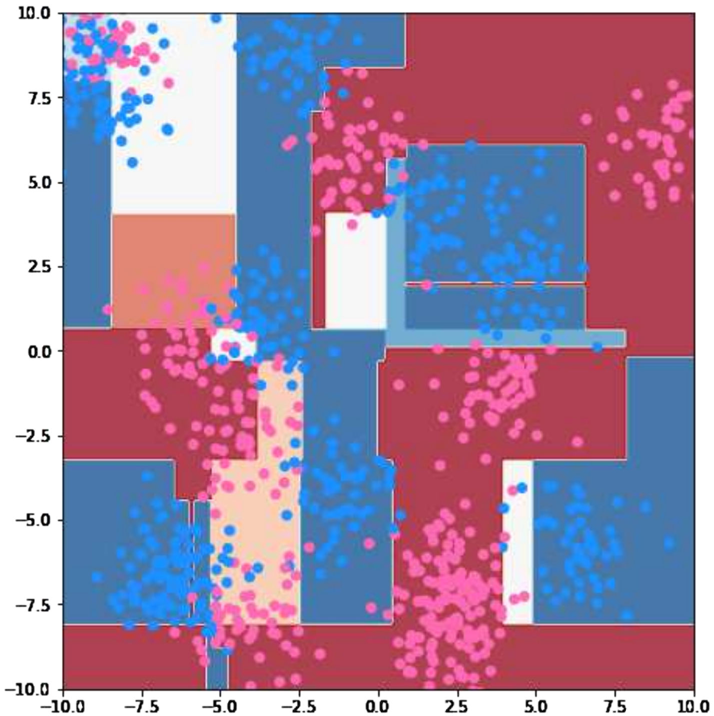
Trees at work

Decision tree, depth=6



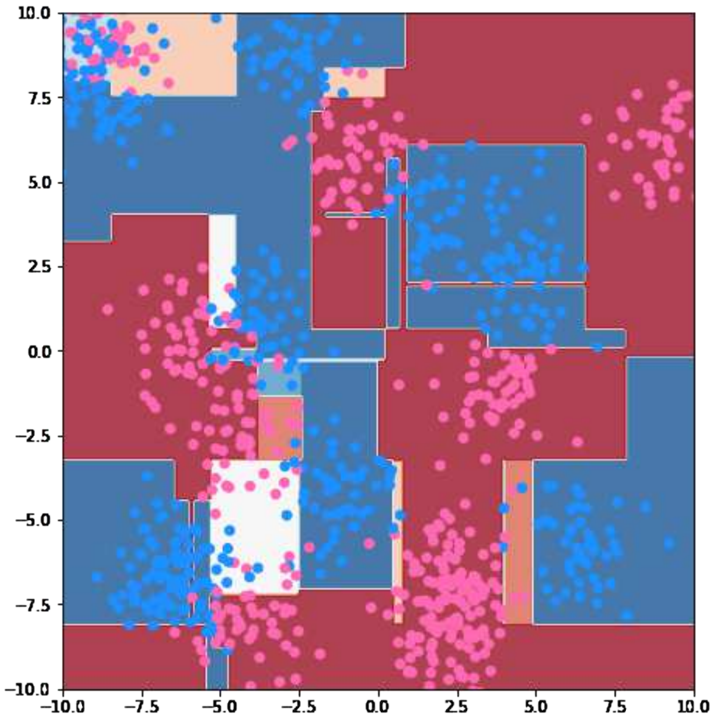
Trees at work

Decision tree, depth=8

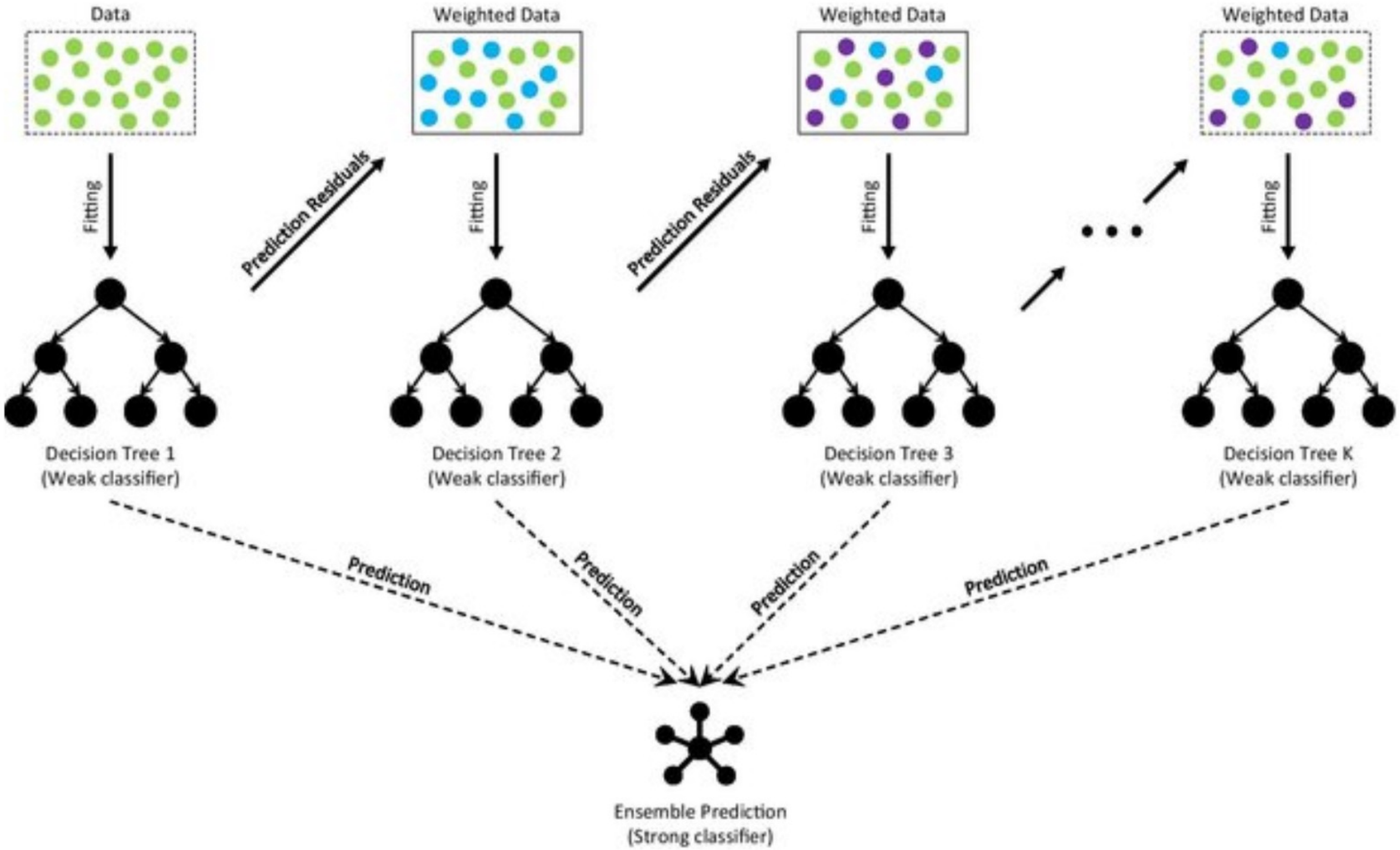


Trees at work

Decision tree, depth=9



Boosted Decision Tree



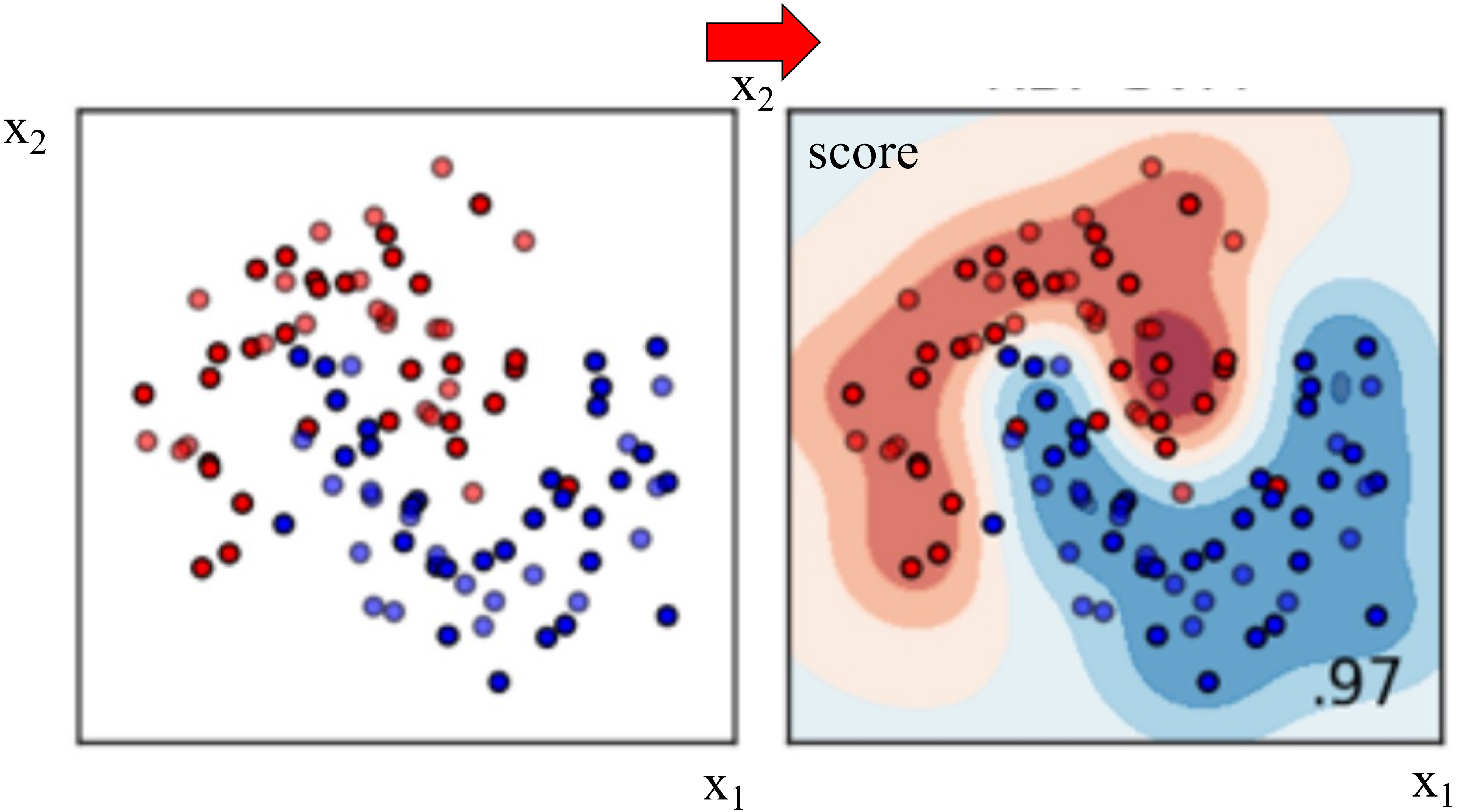
- Gradient Boosted Decision Tree chart

BDT software

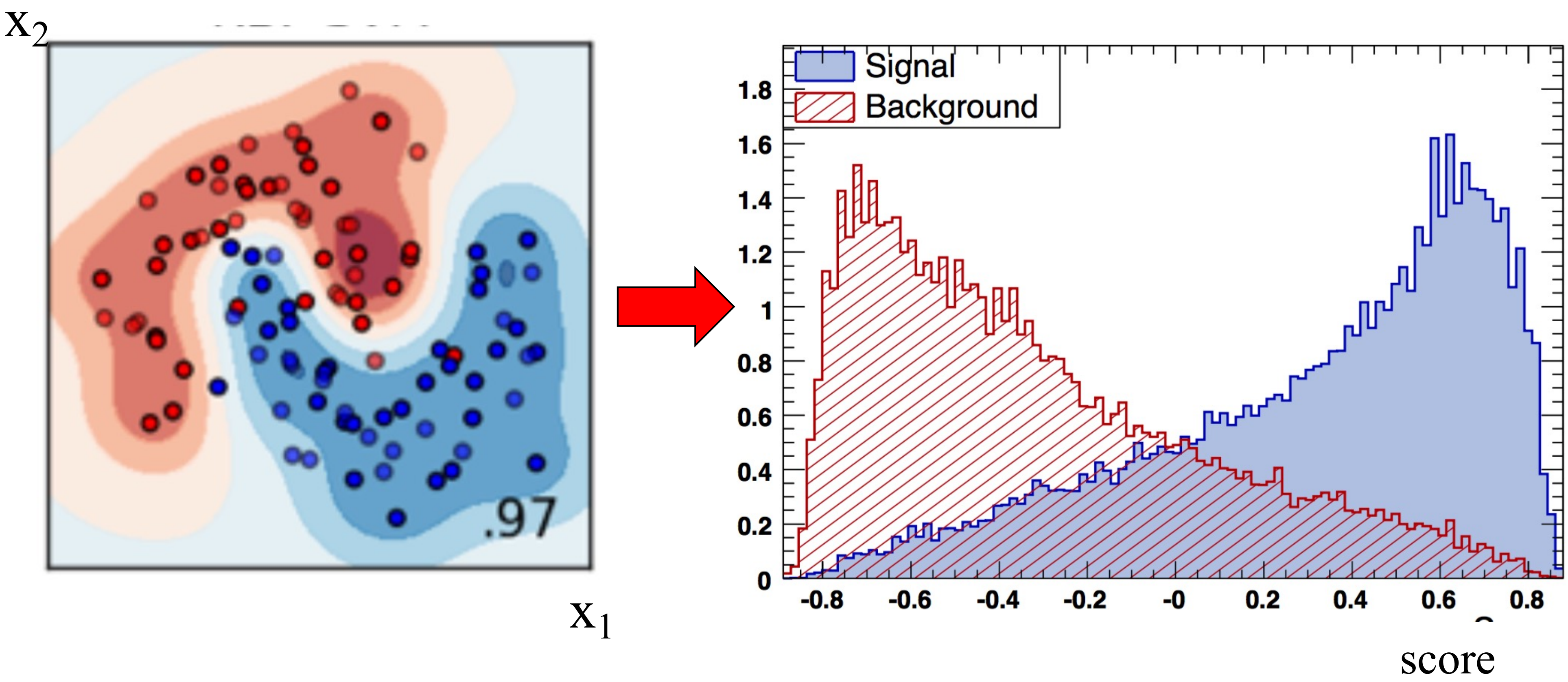
- BDT : first tool of choice for any supervised classification problem with <100 features
- XGBoost (with « hist » option)
- Lightgbm (Microsoft but free open source) (some issue with weighting)
- Sklearn DecisionTreeClassifier

ROC curve, AUC and more

Score

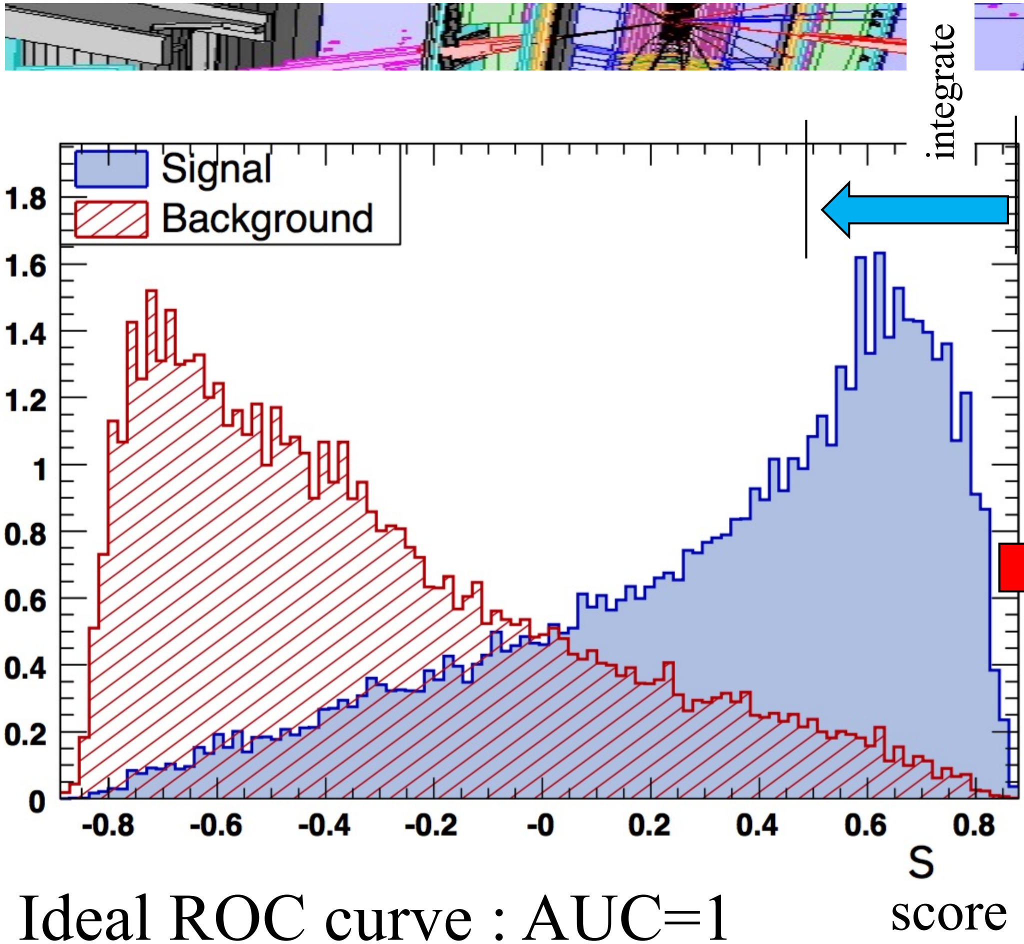


Score

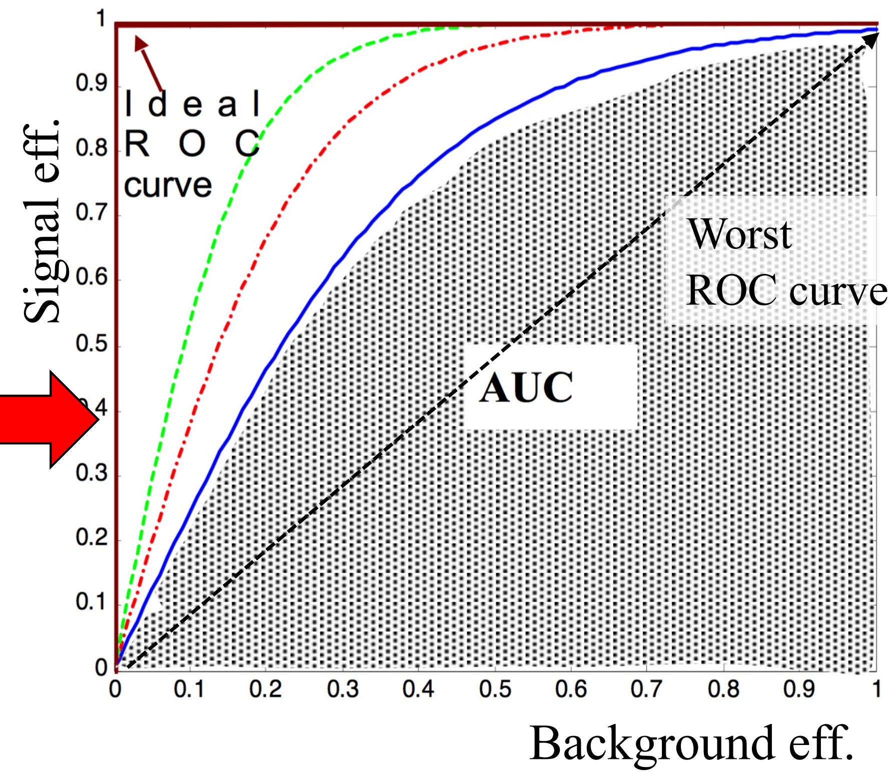


x_2
.97
x_1
Signal
Background
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.8
-0.6
-0.4
-0.2
-0
0.2
0.4
0.6
0.8
score

ROC Curve (BB)



AUC : Area Under the (ROC) Curve



Ideal ROC curve : AUC=1

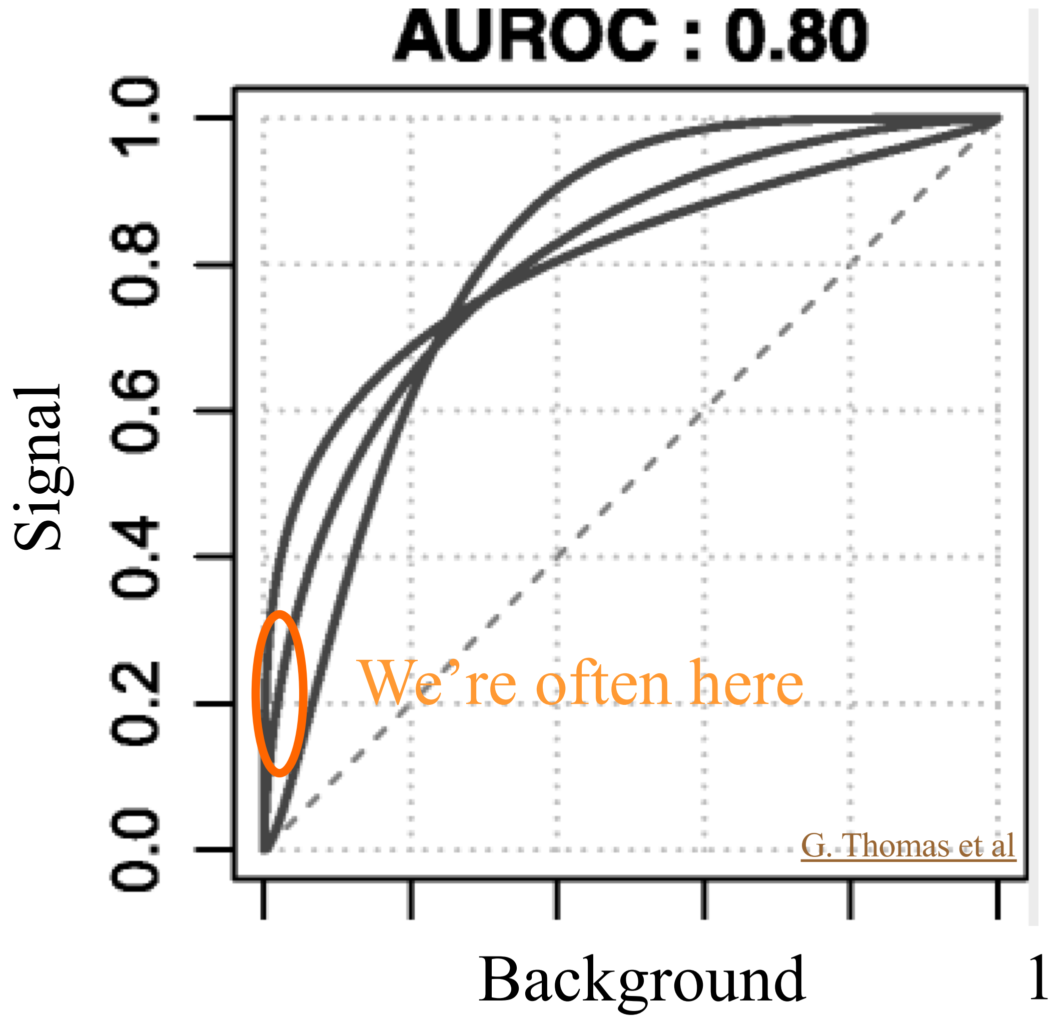
Worst ROC curve : AUC=0.5

AUC<0.5➔bug!

The higher the AUC the better

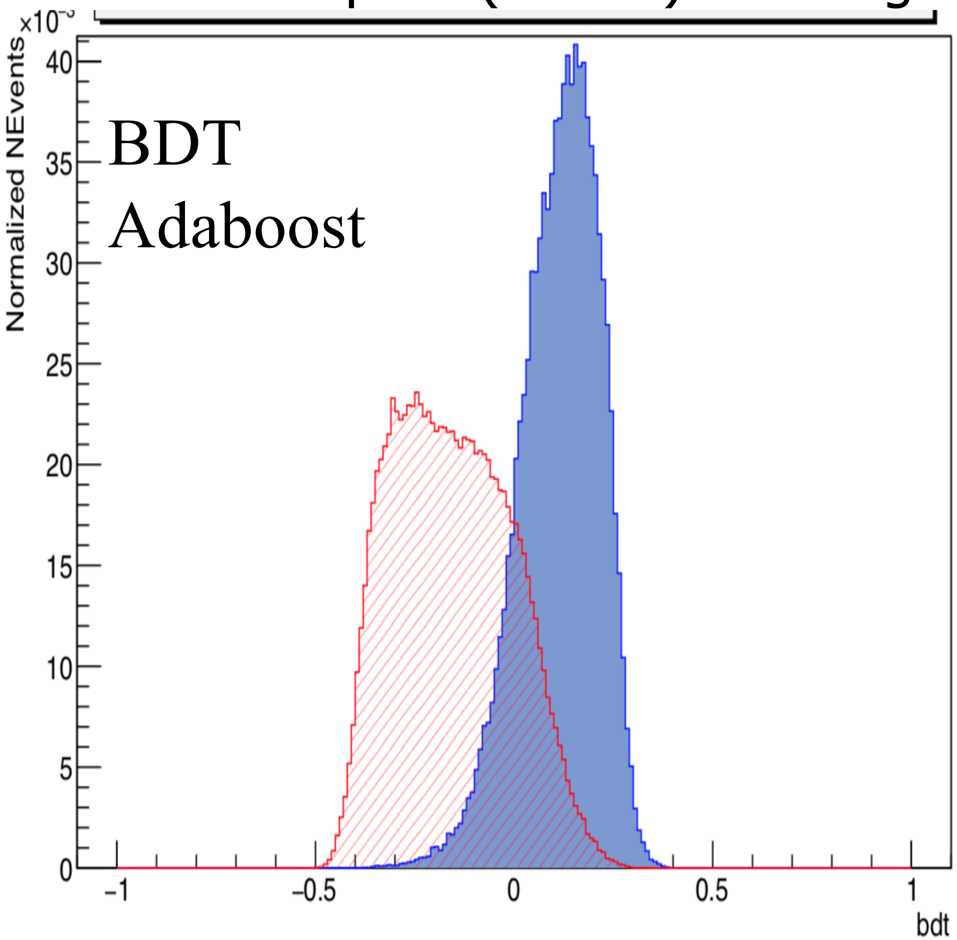
However AUC not the full story

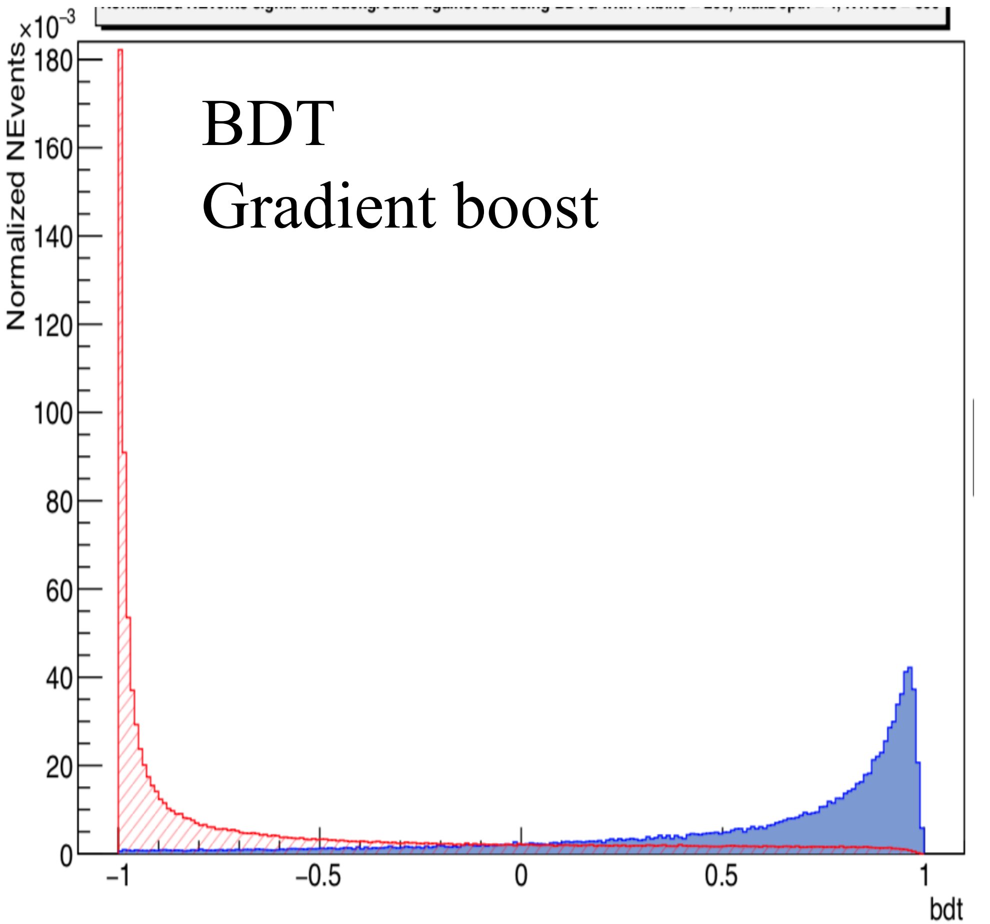
ROC curve pitfall BB



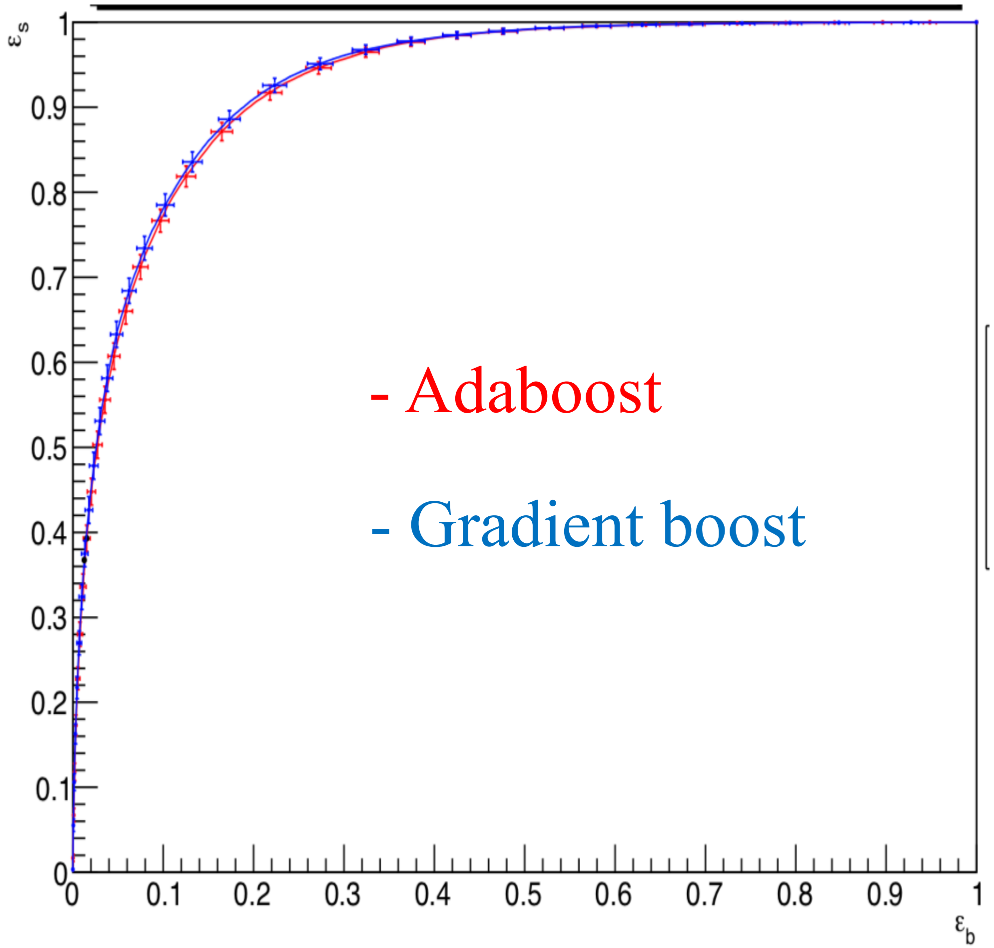
Score

- Score output by algorithm is actually a ranking-variable
  - From most background like to most signal-like
  - Shape is (almost) meaningless

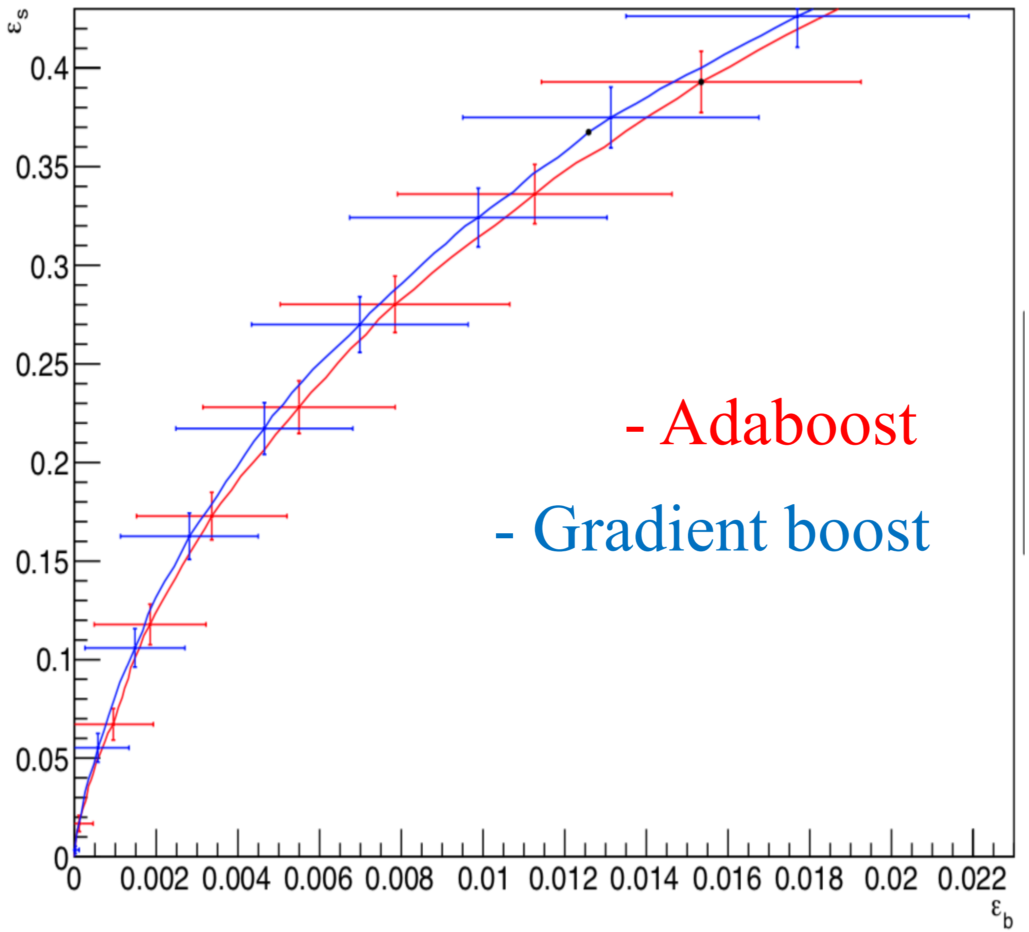




Score (2)

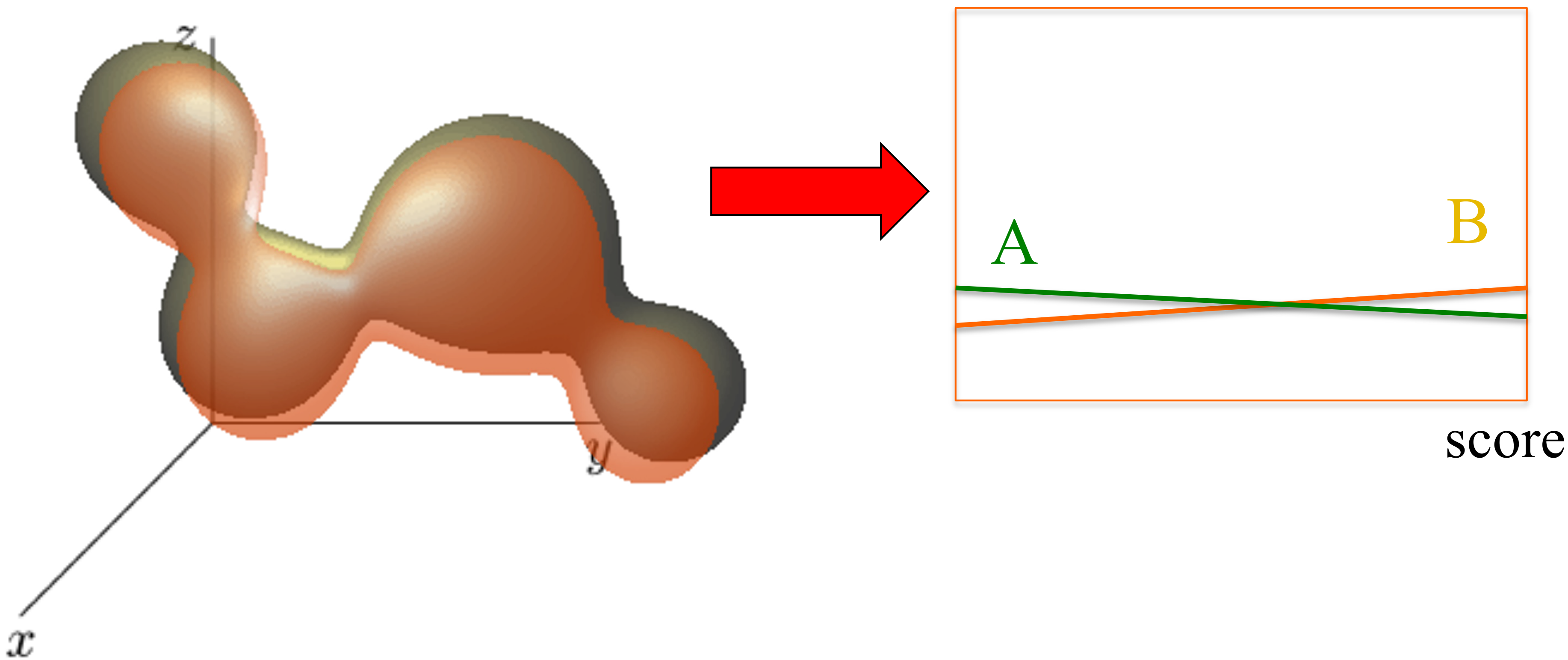


ε_s
ε_b
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
- Adaboost
- Gradient boost



ε_s
ε_b
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
0.022
- Adaboost
- Gradient boost

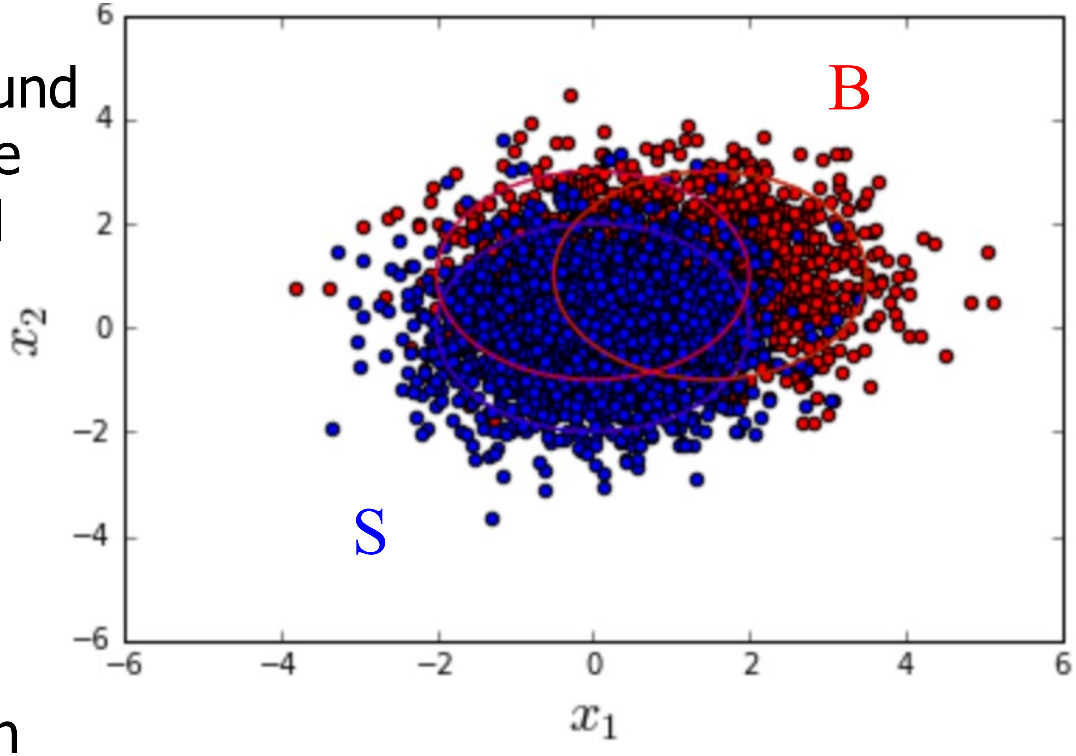
What does a classifier do?



- The classifier "compresses" the two multidimensional "blobs" maximising the difference, without (ideally) any loss of information

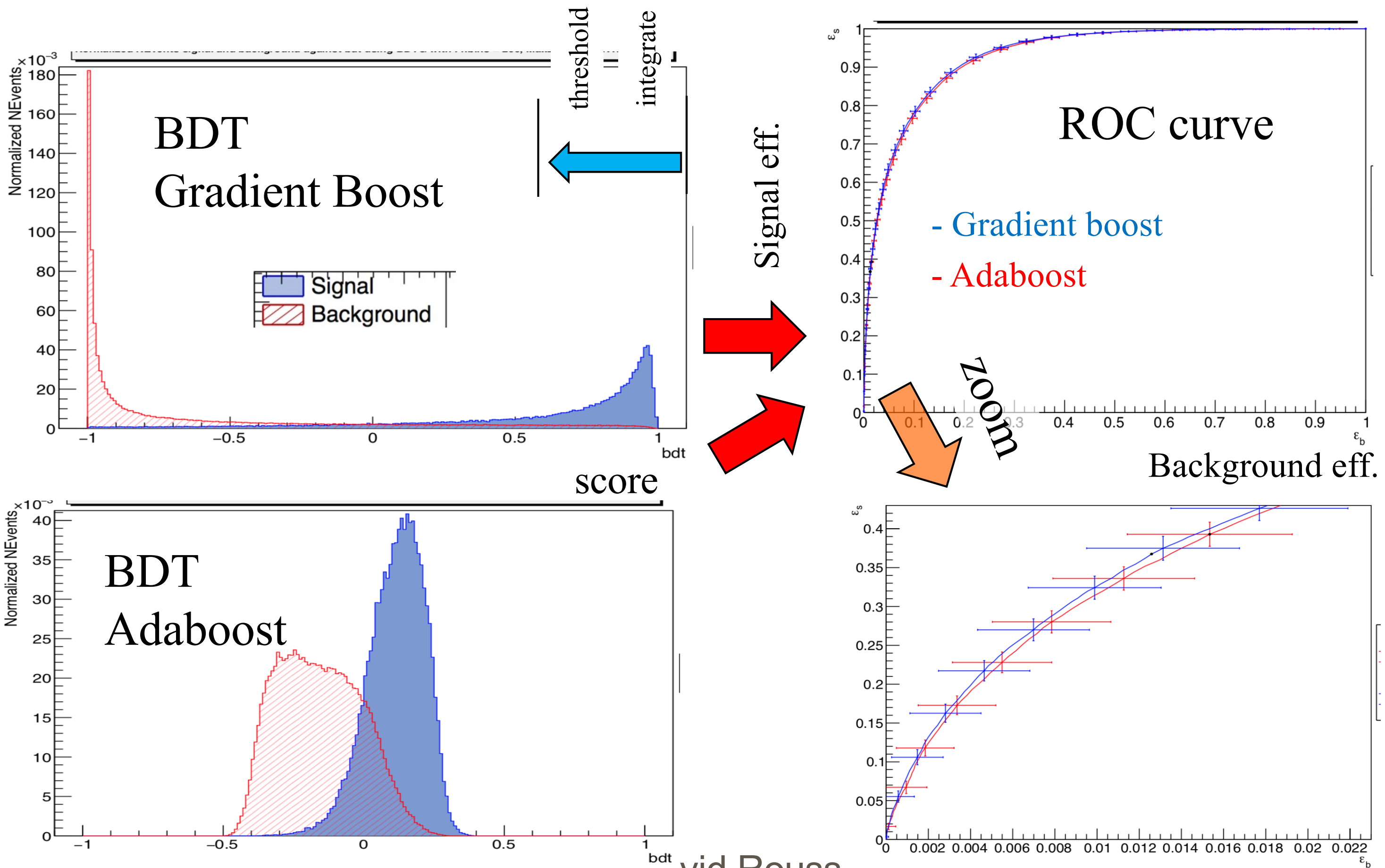
No miracle

- ML (nor Artificial Intelligence) does not do any miracles
- For selecting Signal vs Background and underlying distributions are known, nothing beats likelihood ratio! (often called "Bayesian limit"):
  - $L_S(x)/L_B(x)$
- OK but quite often L_S L_B are unknown
  - + x is n-dimensional
- ML starts to be interesting when there is no proper formalism of the pdf
- ➔mixed approach, if you know something, tell your classifier instead of letting it guess



Significance optimisation

Recall ROC Curve



Significance

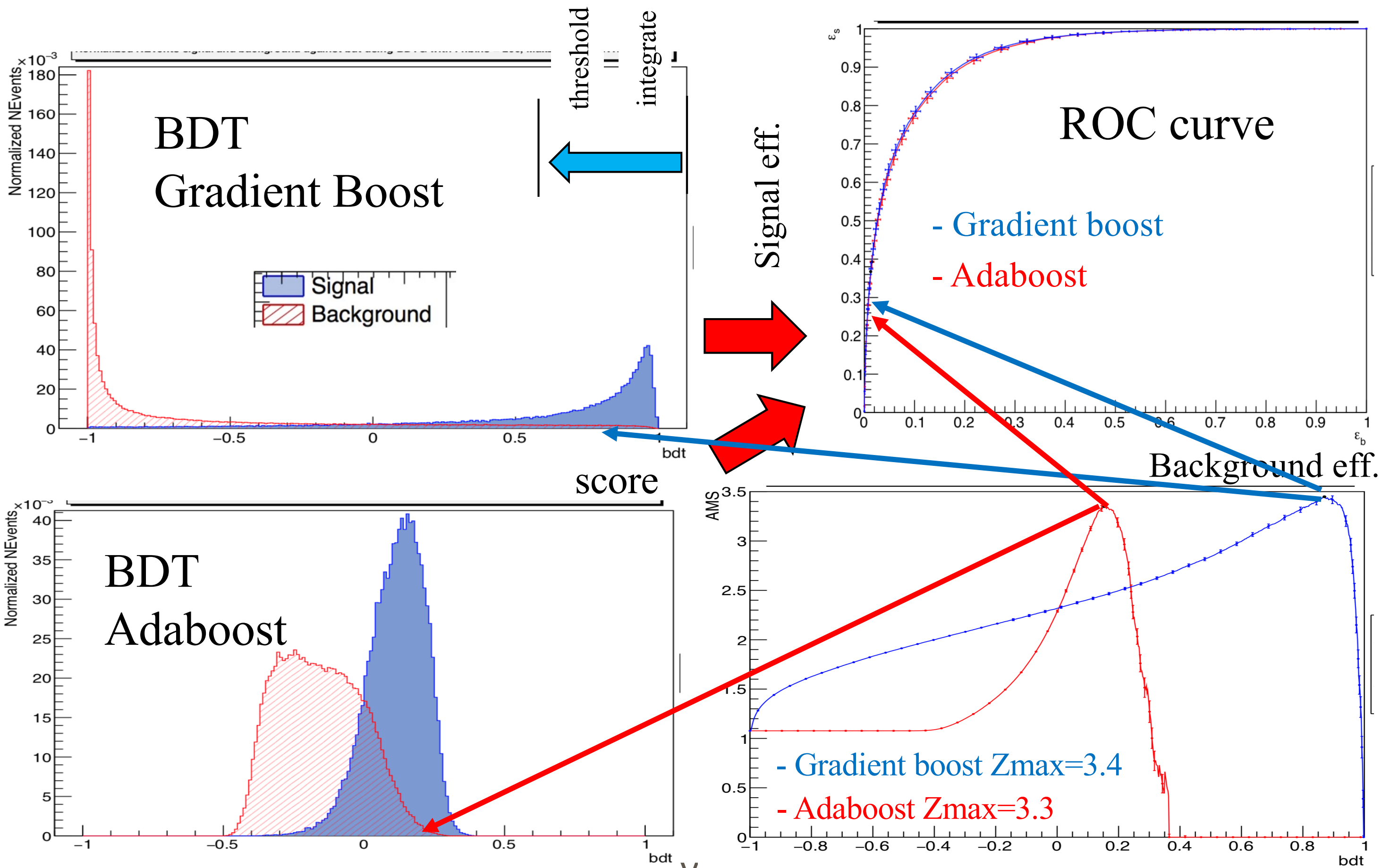
- Imagine counting experiment above threshold

$$s = N_{sig}^{exp} = N_{sig}\epsilon_{sig}$$

$$b = N_{bkg}^{exp} = N_{bkg}\epsilon_{bkg}$$

$$Z = \sqrt{2\left((s+b)\log(1+\frac{s}{b}) - s\right)} \simeq \frac{s}{\sqrt{b}}$$

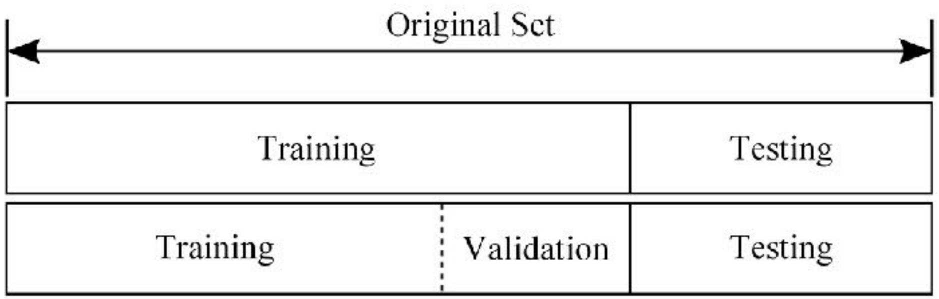
Significance curve



Training, Validation, Test

-

Divide the labelled set into training, validation and testing sets



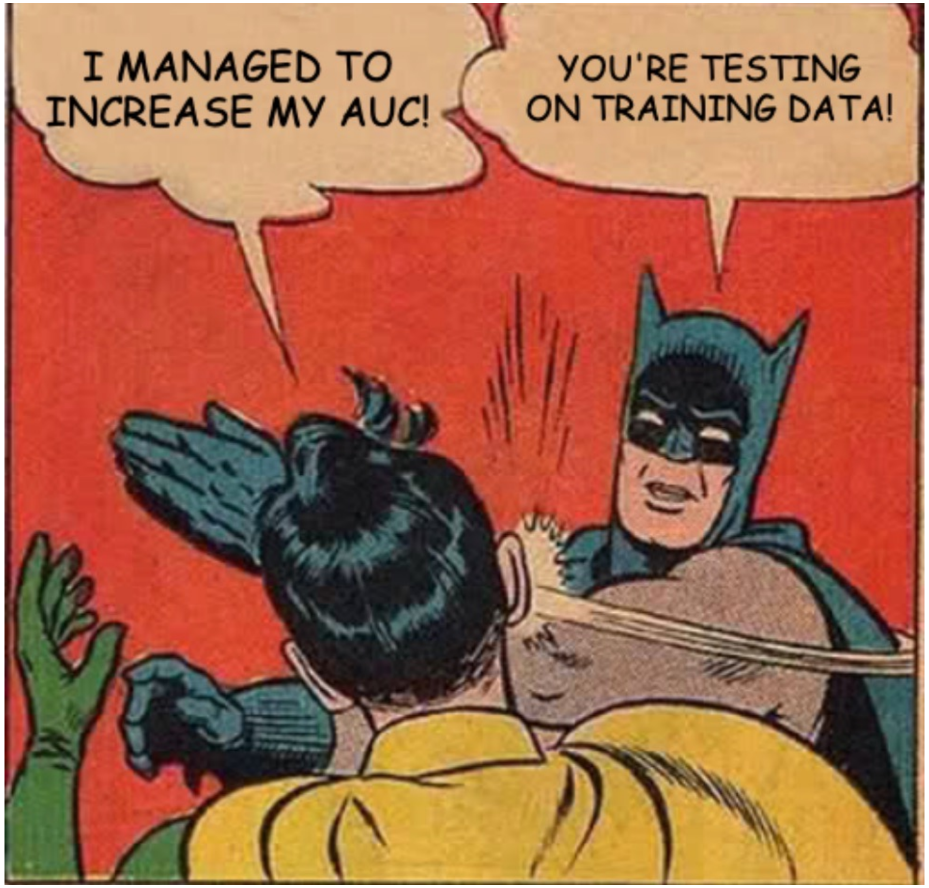
* **<u>Training set:</u> used to train the classifier**
* **<u>Validation set (optional)</u>: choose between different methods, finite-tune parameters,**
* **<u>Testing set:</u> predict the generalization error**

Ideally, look at it only once at the end

<u>No cheat:</u> do not use the test set to train your algorithm!

In practice, we'll only do Train/Test split in the TD

ML highest crime



Horror stories

See page

MIT Technology Review

Also some ChatGPT success stories

Intelligent Machines

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

by Tom Simonite

Jun 4, 2015

The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search company Baidu announced that its

Also in physics…

Training on test set particularly bad because undetectable unless:

- training reproducible
- new i.i.d data

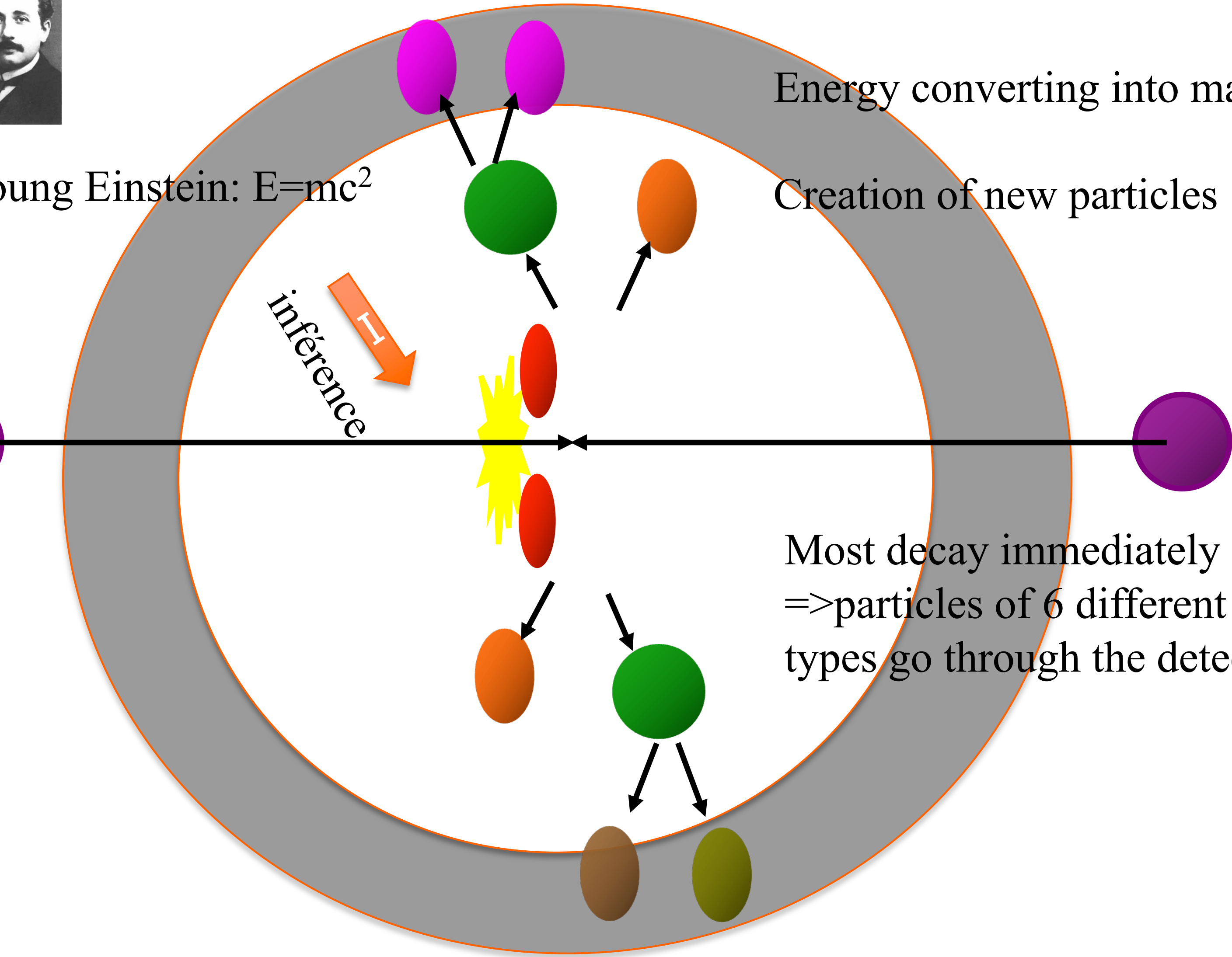
ML for Higgs physics

Using ML to see the Higgs Boson
Using Boosted Decision Tree first
Intro to BDT tutorial

Seeing the Higgs

Proton collisions

Young Einstein: $E=mc^2$



Energy converting into mass

Creation of new particles

Most decay immediately
=>particles of 6 different
types go through the detector

Two fundamental entities

- « Events » :
  - All measurements from one proton collision
  - List of particles with their properties
  - Derived quantities
  - =>ML to help select interesting events « Signal » with respect to « Background »
- « Particles »:
  - Extracted from an event
  - Jet, lepton, photon Missing ET

Before observation, all was known about the Higgs boson, except its mass

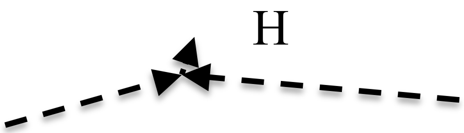
Probabilités de désintégration prédites pour une masse de 125 GeV

H → bb	58%
H → WW*	21%
H → τ+τ-	6.4%
H → ZZ*	2.7%
H → γγ	0.2%

$E=mc^2$

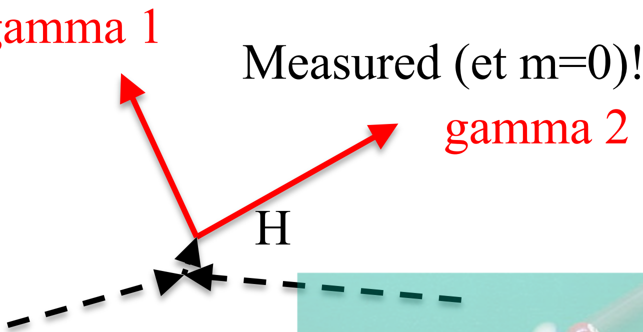
Einstein en 1905

$\boxed{E^2=p^2+m^2}$



H, before decay

$m_H^2=E_H^2-p_H^2$



H,Just after decay

$E_H=E_{g1}+E_{g2}$

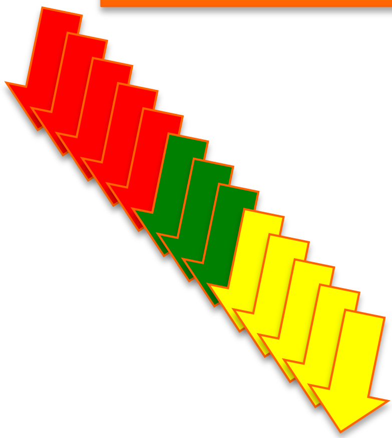
$\vec{p}_H=\vec{p}_{g1}+\vec{p}_{g2}$

⇨we get m_H!

10^{14} collisions / year

Trigger: fast rough selection

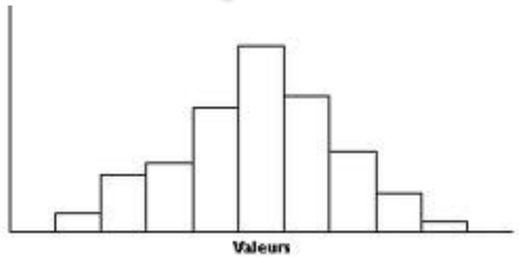
10^{9} events on disk

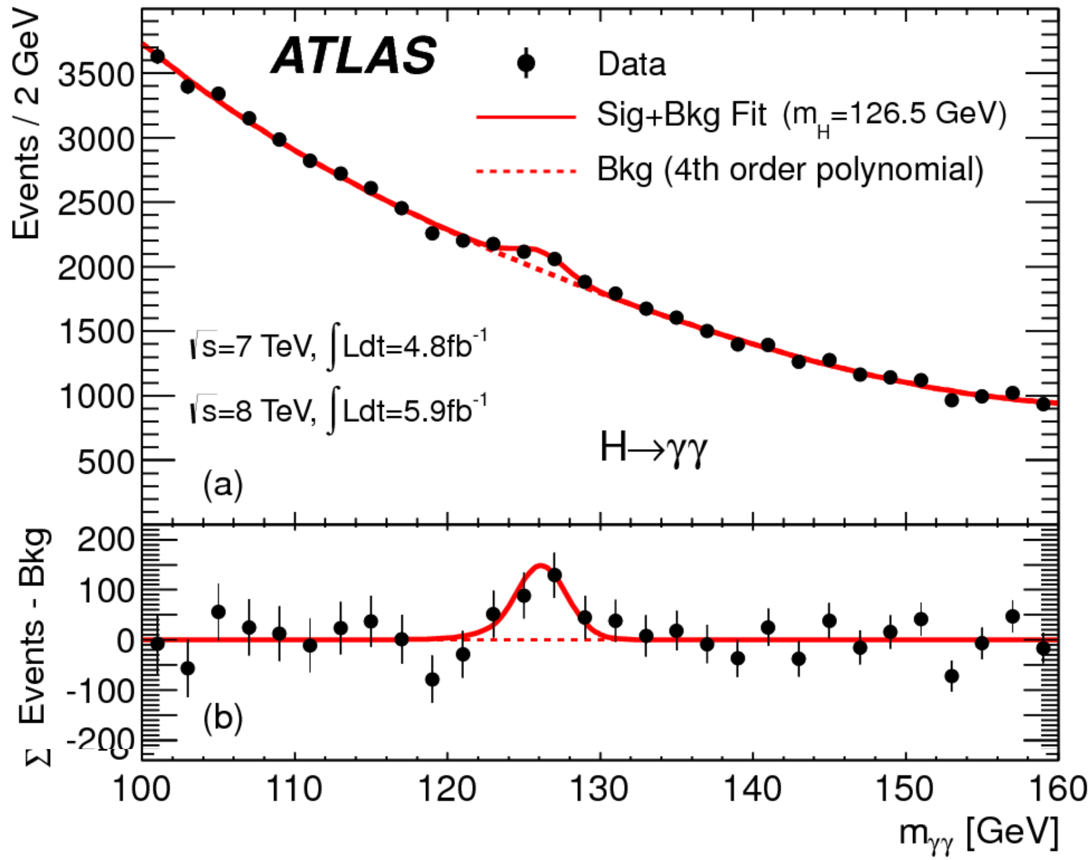


Tri précis

10^{5} events with 2 photons

Mass calculation
→histogramme

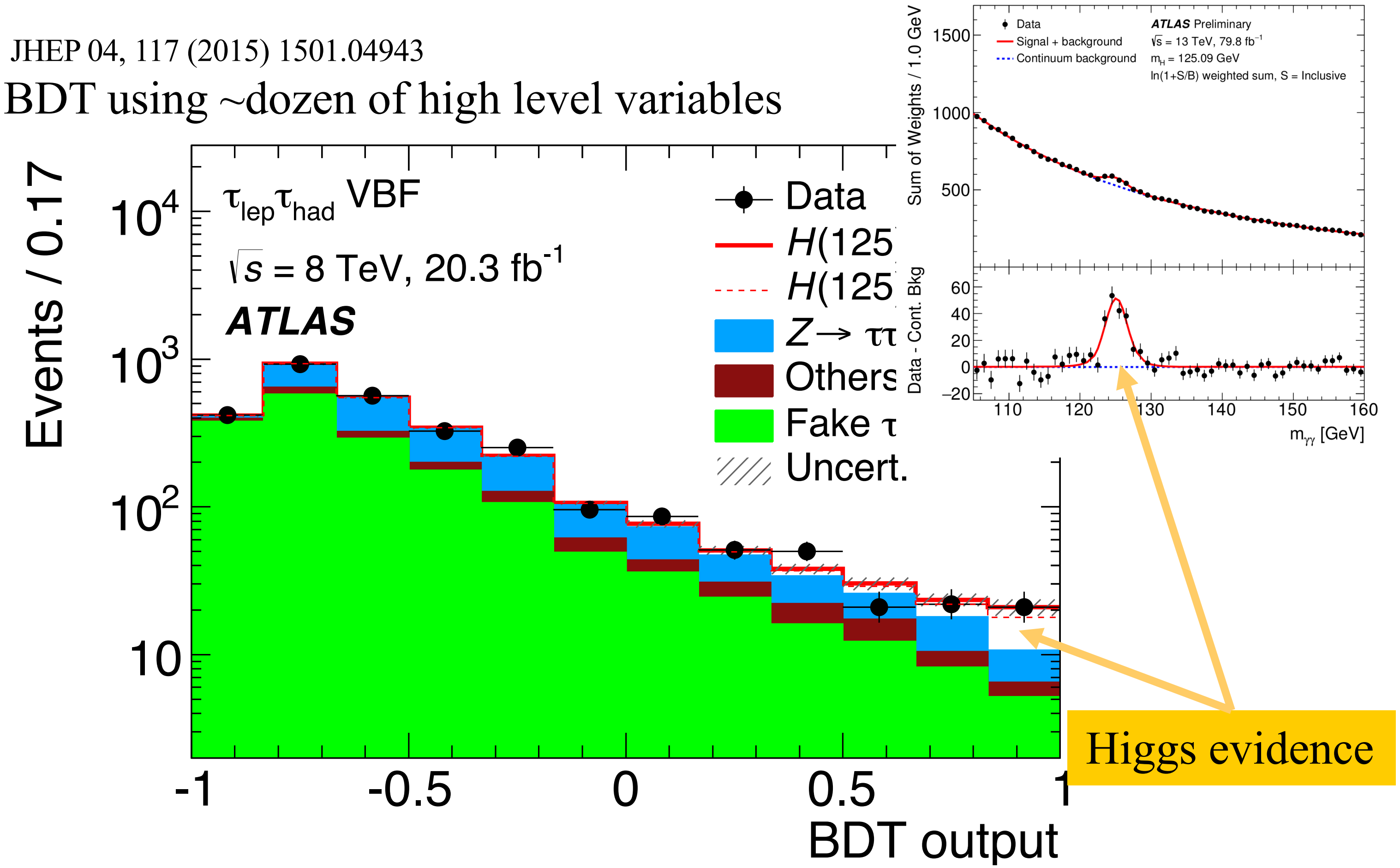



ATLAS
Data
Sig+Bkg Fit (m_H=126.5 GeV)
Bkg (4th order polynomial)
$\sqrt{s}$=7 TeV, $\int Ldt$=4.8fb^{-1}
$\sqrt{s}$=8 TeV, $\int Ldt$=5.9fb^{-1}
$H \rightarrow \gamma\gamma$
(a)
(b)
Events / 2 GeV
Σ Events - Bkg
$m_{\gamma\gamma}$ [GeV]
100
110
120
130
140
150
160
500
1000
1500
2000
2500
3000
3500
200
100
0
-100
-200

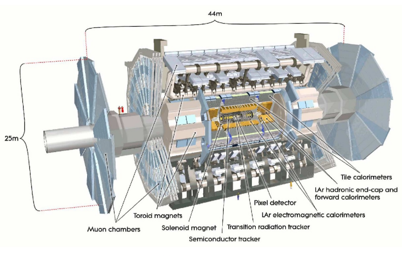
Classifier in Higgs Physics

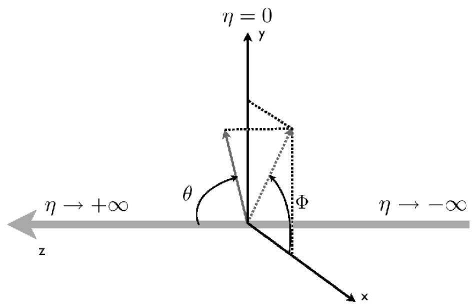
JHEP 04, 117 (2015) 1501.04943

BDT using ~dozen of high level variables



Coordinates





- P : momentum
- E : energy =sqrt(P^2+M^2)~P because P>>M
- Angles (cylindrical)
  - o ϕ : azimuth angle]-π, +π]
  - o θ : dip angle [0, +π]
  - o η : eta, pseudo-rapidity = -log(tan(θ/2)), ~[-5,5]
- P_T : =P sin(θ) : transverse momentum
- ME_T : Missing Transverse Energy = $-\Sigma_{\text{all particles}} \vec{P}_T$: estimator of transverse momentum of neutrinos

H→WW

- One of the Higgs Discovery channel
- $H \rightarrow W^+(\rightarrow l^+ \nu)\ W^-(\rightarrow l^- \nu)$
  - ➔2 leptons of opposite charge
  - Neutrinos undetected ! => Missing Transverse Energy
  - No invariant mass peak!
- Background :
  - Other processes leading to $W^+(\rightarrow l^+ \nu)\ W^-(\rightarrow l^- \nu)$

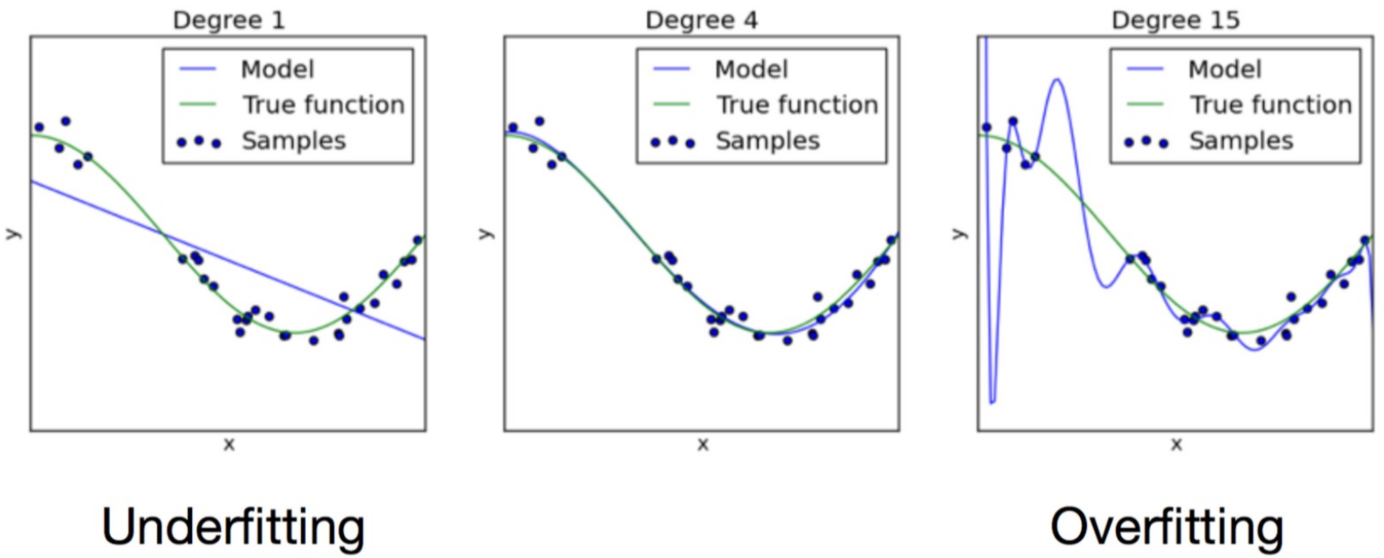
tutorial

- Open Google Colab
- Search for my github repository: https://github.com/dhrou/HEPMLtutorials
- Open HEPML_HandsOn_BDT.ipynb
- Remember to save a copy first
- You can also run locally by switching COLAB=False, and downloading the dataset and fixing the path (under « Load Events »)

Under/Over-training

What is Overfitting

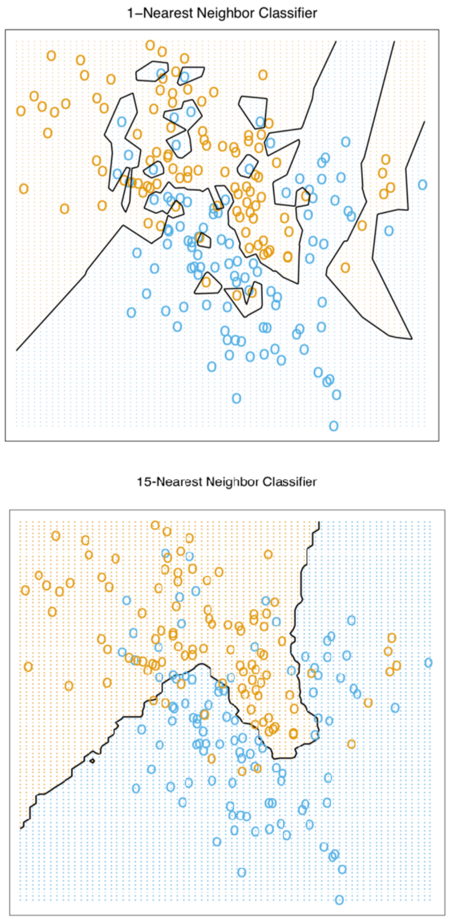
45

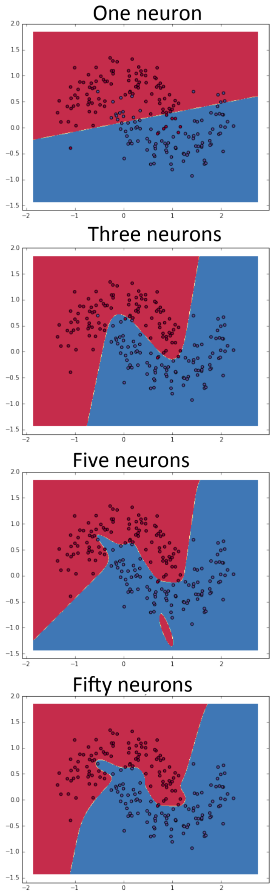


http://scikit-learn.org/

- What models allow us to do is **generalize** from data
- Different models generalize in different ways

Overtraining examples



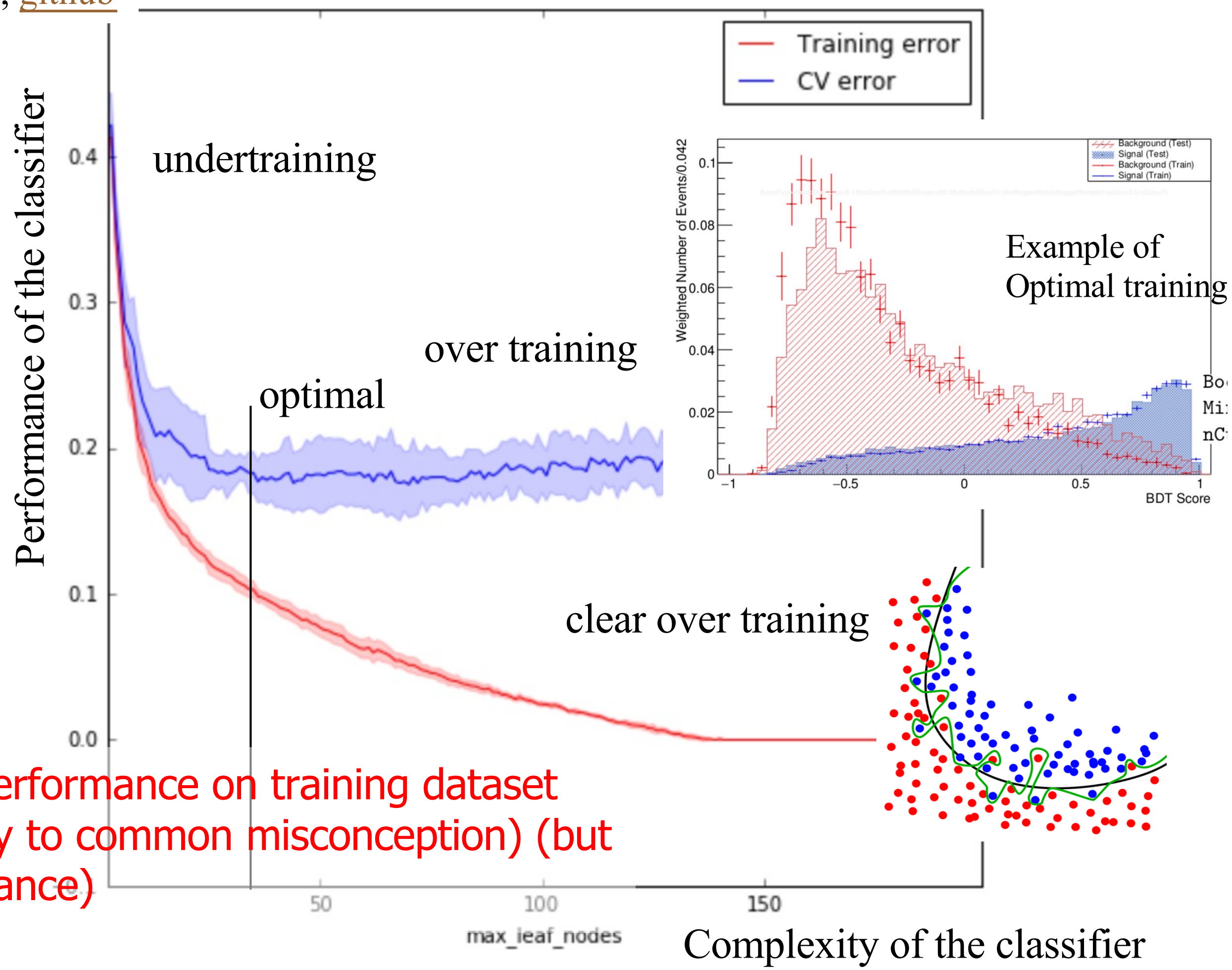


http://www.wildml.com/2015/09/impl

- Overtraining affect all algorithms
- …when model is too complex wrt amount of training data

under/over training

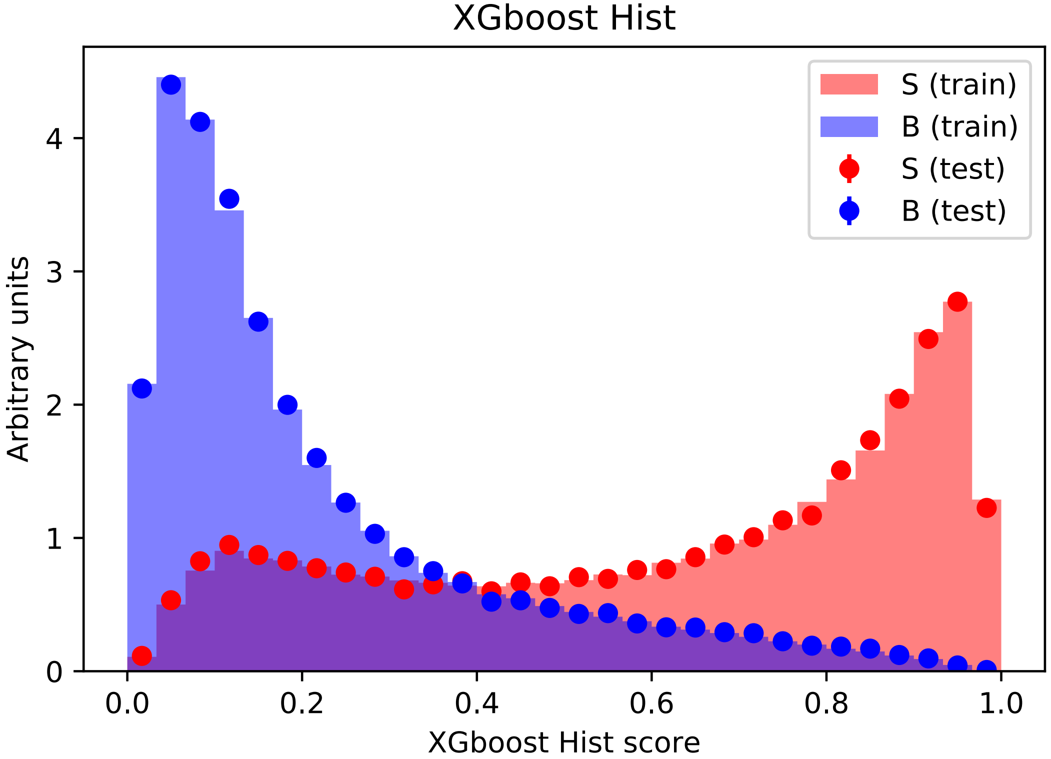
Gilles Louppe, github



It is OK to have better performance on training dataset wrt test dataset (contrary to common misconception) (but it is not the real performance)

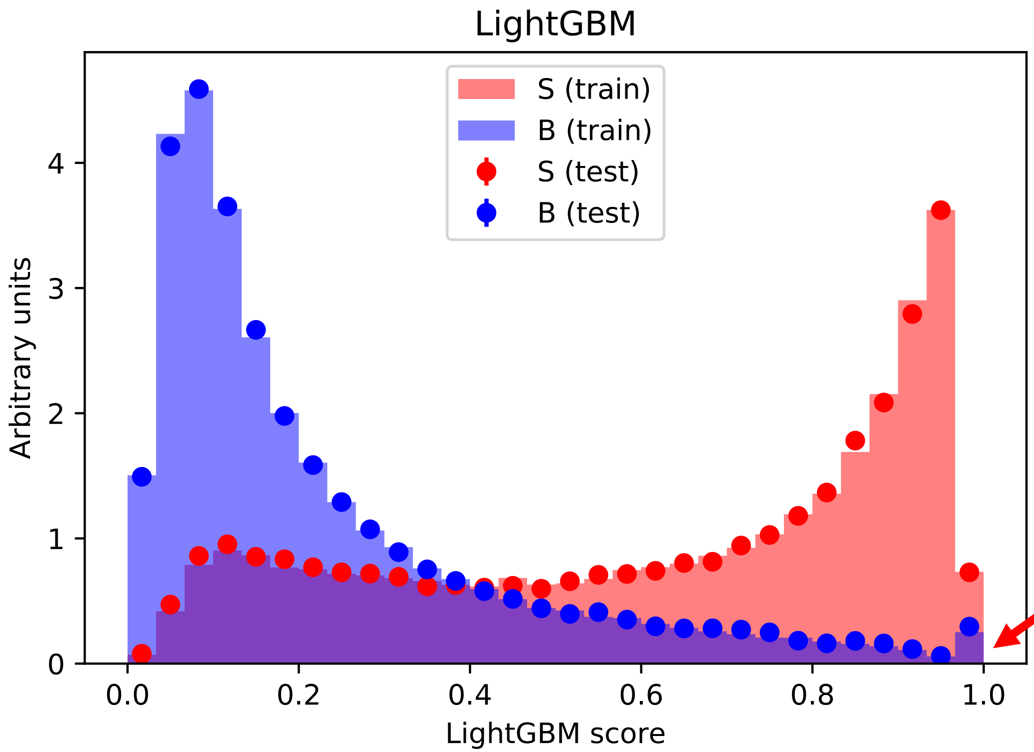
Training vs test check

- The score distribution should match…
- …some discrepancy is OK



Training vs test check (2)

- Important to check



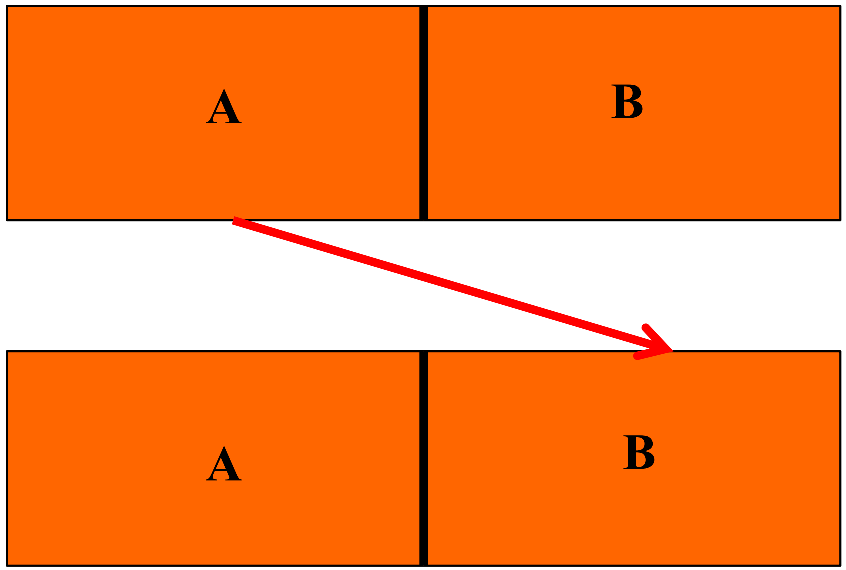
Bug!

Cross-validation

Cross-Validation

One-fold Cross Validation

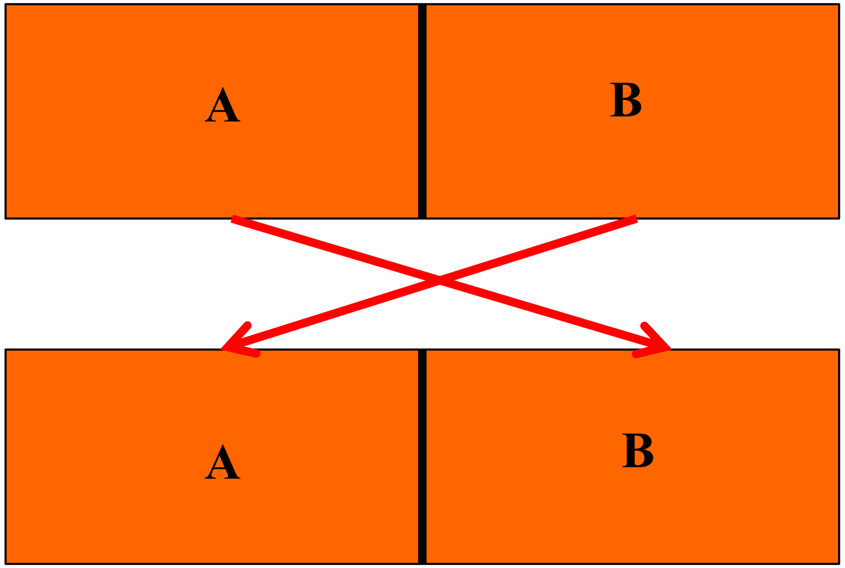
Goal of CV is to measure performance and optimise hyper-parameters



Standard basic way

Cross-Validation

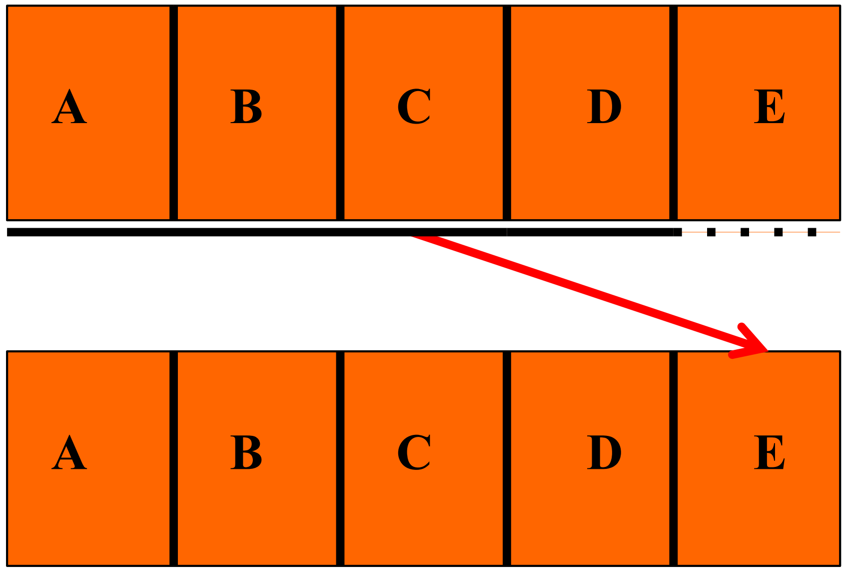
Two-fold Cross Validation



➔test statistics = total statistics
➔double test statistics wrt one fold CV
➔(double training time of course)

Cross-Validation

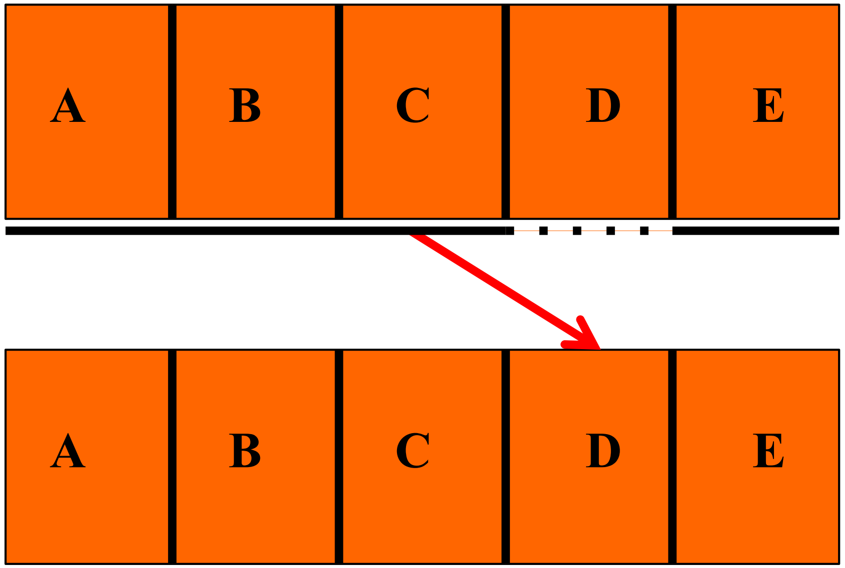
5-fold Cross Validation



same test statistics wrt two-fold CV,
larger training statistics 4/5 over ½ (larger training time as well)

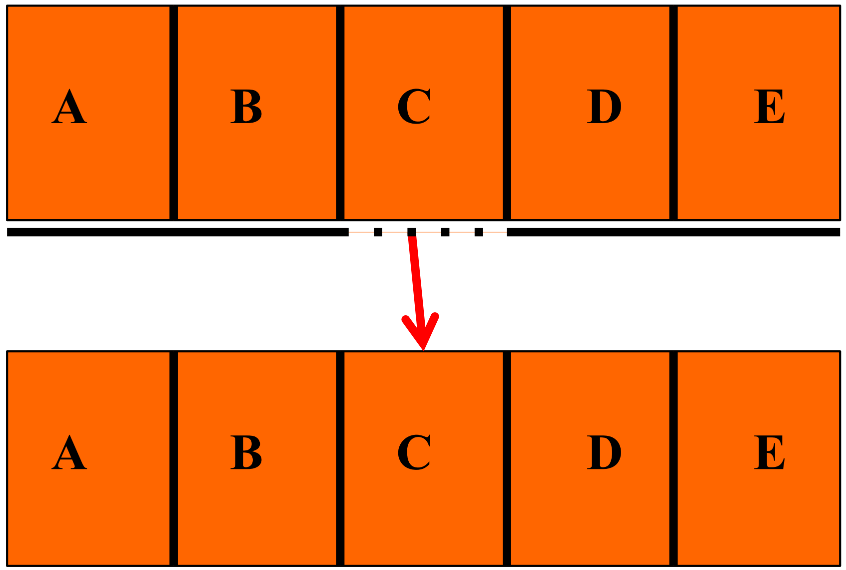
Cross-Validation

5-fold Cross Validation



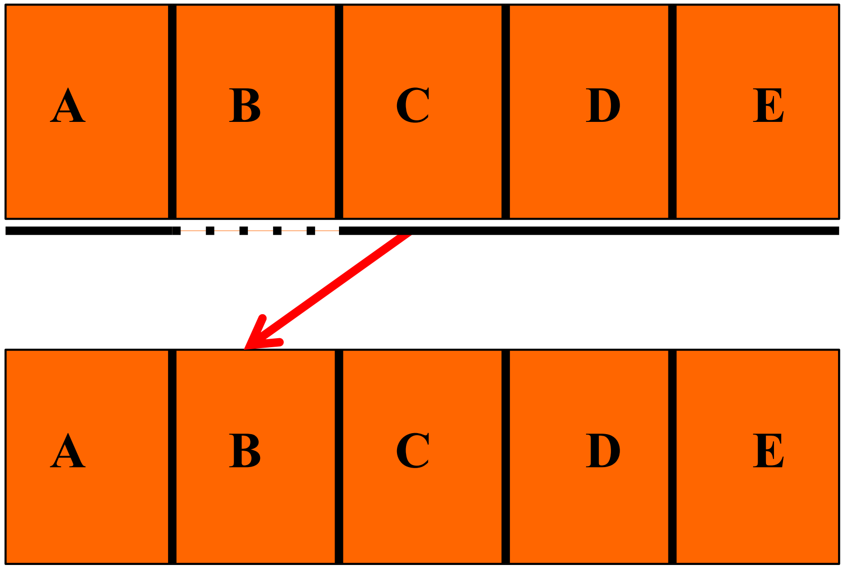
Cross-Validation

5-fold Cross Validation



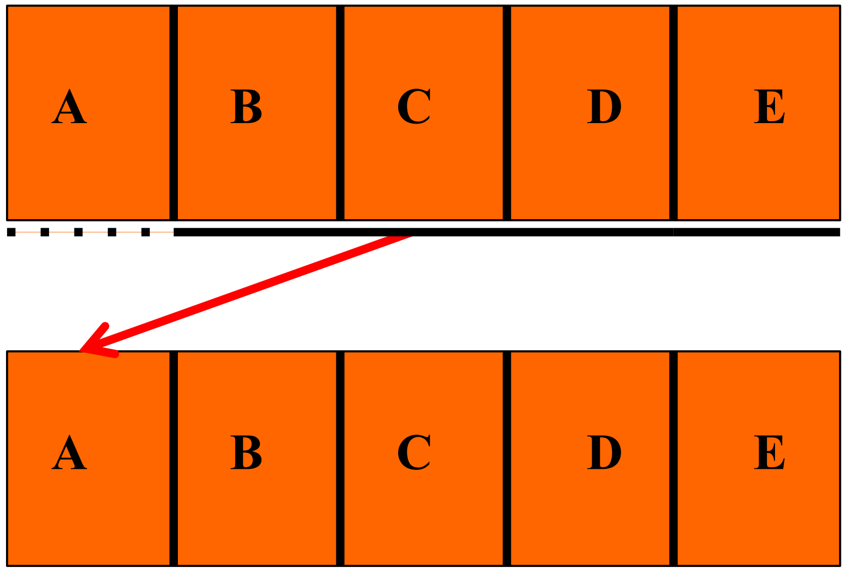
Cross-Validation

5-fold Cross Validation



Cross-Validation

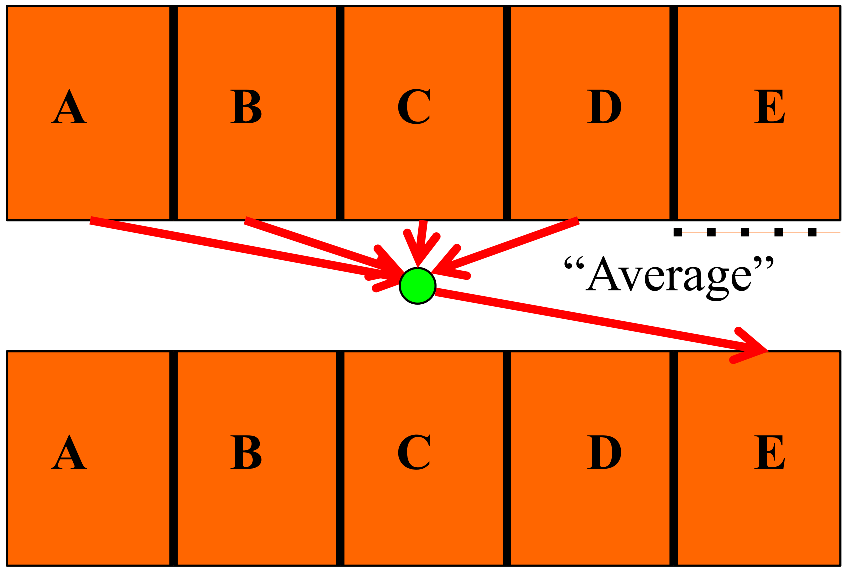
5-fold Cross Validation



Note : if hyper-parameter tuning, need a third level of independent sample “nested CV”

Cross-Validation

5-fold Cross Validation “à la Gabor”



Average of the scores on A B C D is
often better than the score of one training ABCD
bonus: variance of the samples an estimate of the statistical uncertainty
(also save on training time)

Cross-validation

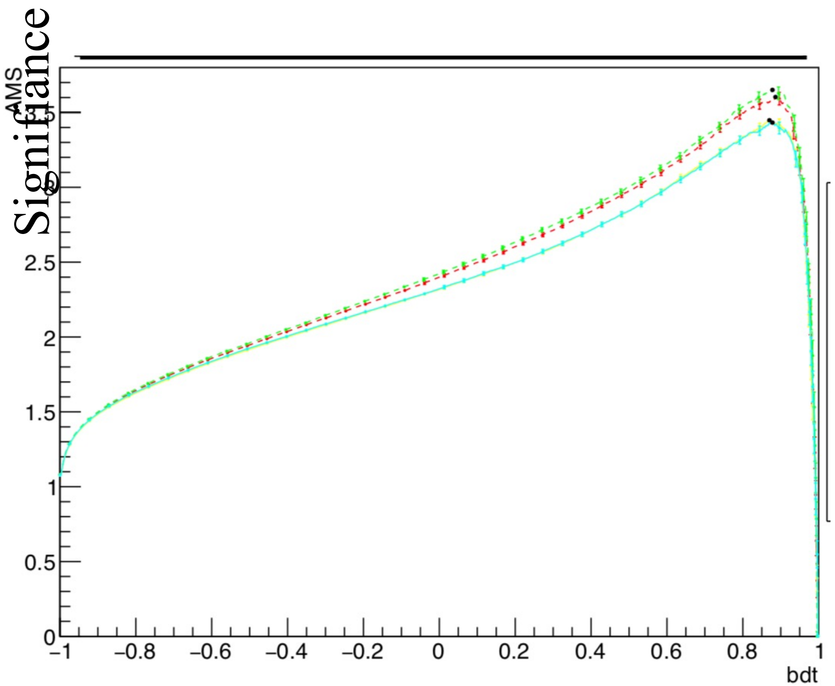
Train on …

Test on …	A	B	C	D	E	
A	X					➔Testing variance
B		X				➔Testing variance
C			X			➔Testing variance
D				X		➔Testing variance
E					X	➔Testing variance
	➔Training variance	➔Training variance	➔Training variance	➔Training variance	➔Training variance	

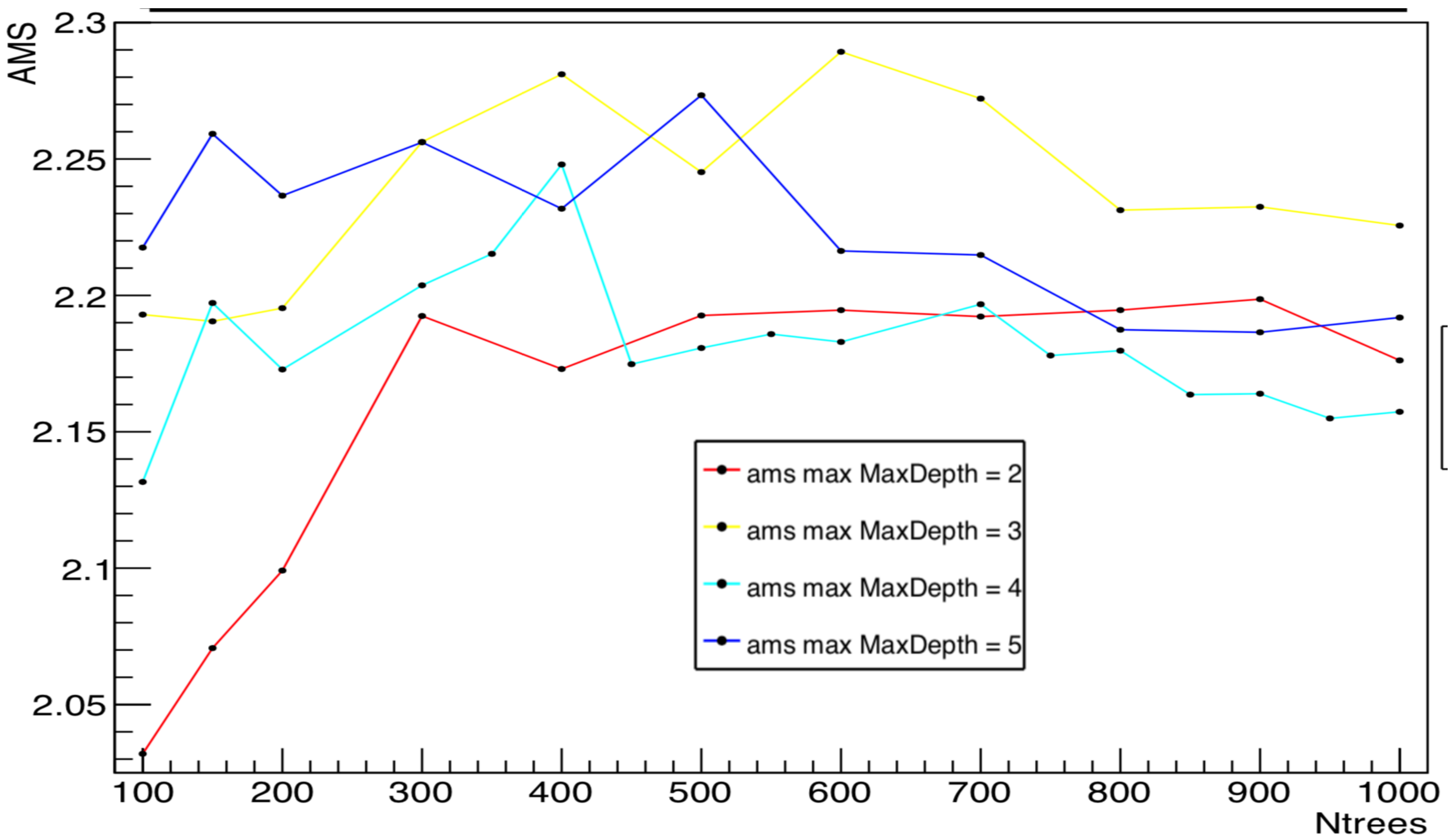
Hyper-Parameter Optimisation (HPO)

Hyper Parameters optimisation

- HP : all the auxiliary parameters of the algorithm, like Depth of the tree, Number of trees etc….
- HPO : optimising these parameters
- Very algorithm dependent
- A few are usually important



HPO example



AMS
2.3
2.25
2.2
2.15
2.1
2.05
100
200
300
400
500
600
700
800
900
1000
Ntrees
ams max MaxDepth = 2
ams max MaxDepth = 3
ams max MaxDepth = 4
ams max MaxDepth = 5

Weights and multi-dimensional reweighting

Absolute normalisation

- Say you are doing an experiment at the LHC
- You are looking for a particular type of event
- How many do you expect ?
- $N^{prod}=L * \sigma(\theta)$
  - N^{prod}= number of produced events (before detector effect)
  - L « integrated luminosity » : for example 138 fb^{-1} for LHC data taking at 13TeV center of mass energy in 2015-2018 prop number of proton collisions
    - 1 barn is 10^{-28} m^2
    - proportional to the total number of proton collision
  - σ (θ) : cross-section (in barn), can be calculated from first principles and θ parameters from nature (electric charge, higgs boson mass etc…)
- $N^{exp}=L * \sigma(\theta)*\varepsilon$
  - N^{exp}= number of expected events (actually counted in the detector). N^{exp} is a real number. The actual number of observed event will follow Poisson (N^{exp})
  - ε : efficiency, probability to detect a produced event (1. if perfect detector).
    - Measured on simulation (calibrated on data)
    - Can be product of many terms like: ε trigger $*$ ε acceptance $*$ ε lepton $*$ ….

Simple Event Counting Experiment

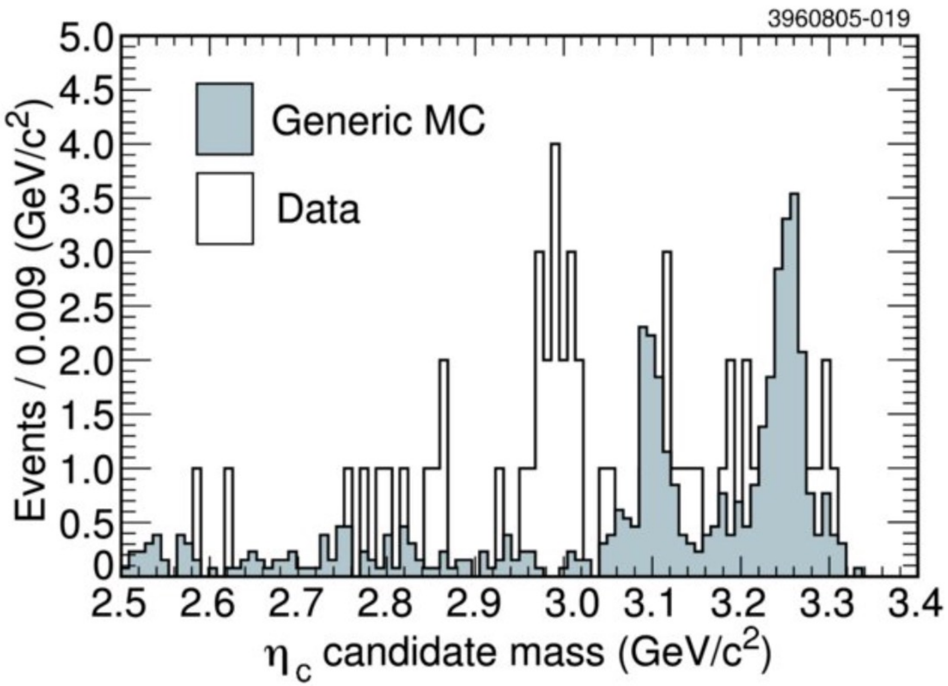
- One signal, we have some estimate of $\sigma_{sig}(\theta)$ but we actually want to assess its existence (exp==expected)
  - N^{exp}_{sig}==s=L * σ_{sig}*ε_{sig}
- one well-known background :
  - N^{exp}_{bkg}==b=L * σ_{bkg}*ε_{bkg}
- N^{exp}=s+b
- We do the experiment and count N^{obs} events
- Hence we measure:
  - $\sigma_{sig}=(N^{obs}- b)/(L*\varepsilon_{sig})$
  - $\sigma_{sig}=(N^{obs}- L*\sigma_{bkg}*\varepsilon_{bkg})/(L*\varepsilon_{sig})$
- Key inputs : ε_{sig} ε_{bkg} determined from simulated datasets

Weights for overall normalisation

- $b=L * \sigma_{bkg} * \varepsilon_{bkg}$
- We measure on simulation : $\varepsilon_{bkg} = N_{bkg\ pass}/N_{bkg\ total}$
  - with $N_{bkg\ pass}$, number of events passing some criteria e.g. momentum of the two photons greater than 25 GeV, BDT score above 0.8 etc…
  - So $b= L * \sigma_{bkg} * N_{bkg\ pass}/N_{bkg\ total}$
- We can define an event weight : $w_i = L * \sigma_{bkg} / N_{bkg\ total}$
- And then simply: $b=\Sigma_{pass}\ w_i$
- Beware : if I take an unbiased subset of x% of dataset, I need to scale the weights by 1/x, so that
- $b^{subset}=\Sigma^{subset}{}_{pass}\ w^{subset}{}_i=(1/x) * \Sigma^{subset}{}_{pass}\ w_i \sim b$

Data / MC histo comparison

- Then one can histogram directly any quantity (using the weights) and it is normalised correctly to the real data
- By convention, real data is almost never weighted

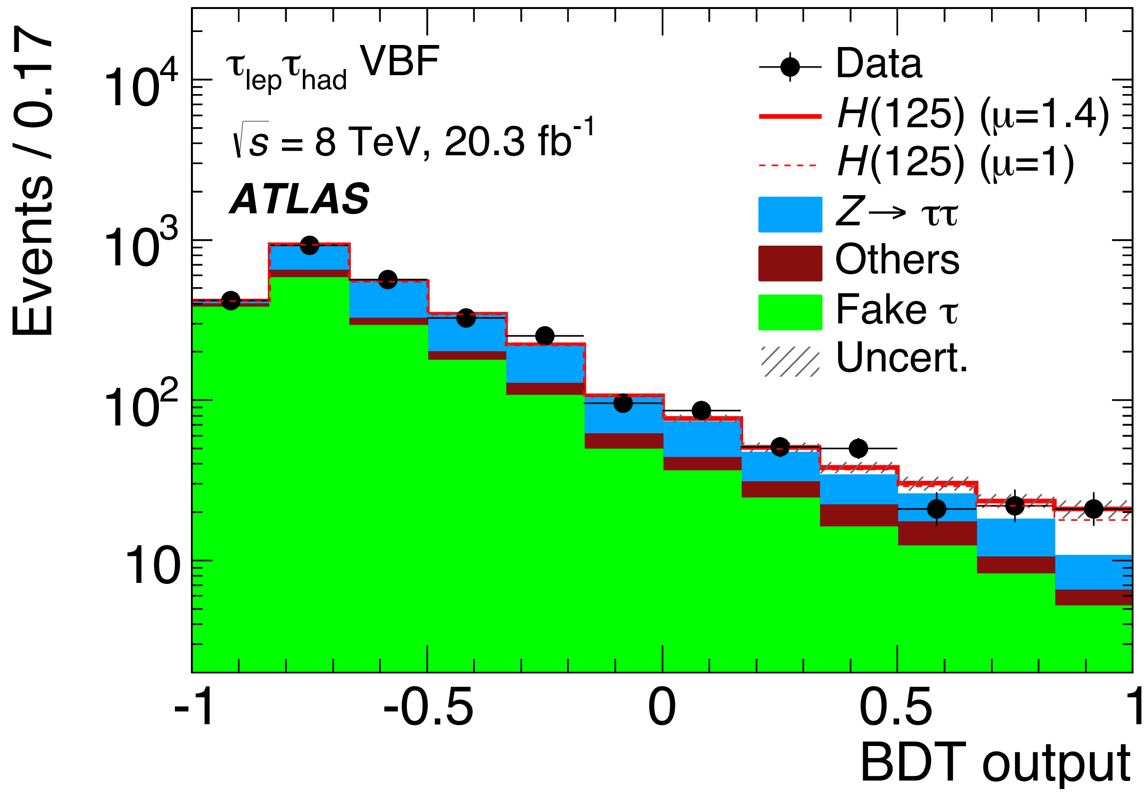


Case of multiple backgrounds

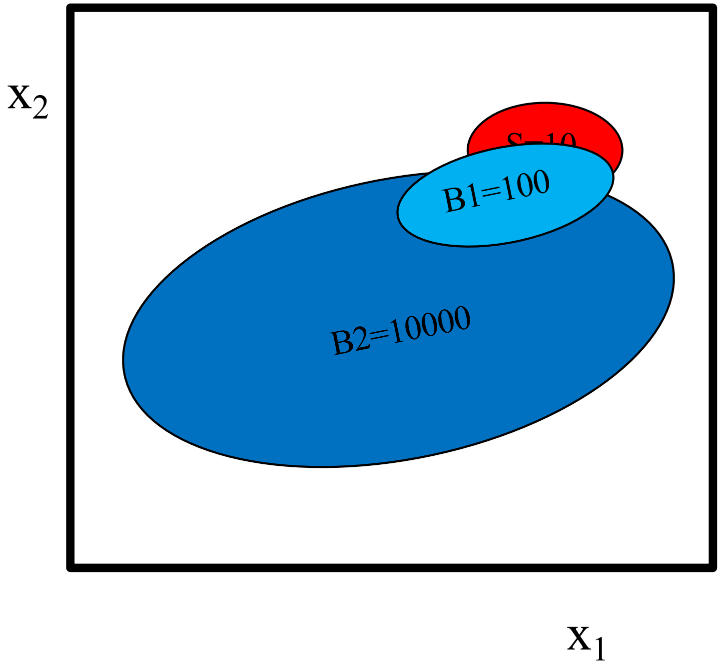
- Now suppose we have two different backgrounds:
- $b=b_1+b_2=L * \sigma_{bkg1}*\varepsilon_{bkg1} +L * \sigma_{bkg2}*\varepsilon_{bkg2}$
- $b=b_1+b_2=L * \sigma_{bkg1}* N_{pass1}/N_{total1} +L * \sigma_{bkg2}* N_{pass2}/N_{total2}$
- If I define the event weight
  - For dataset bkg 1 : $w_i= L * \sigma_{bkg1/} N_{total1}$
  - For dataset bkg 2 : $w_i= L * \sigma_{bkg2/} N_{total2}$
- And then : $b=\Sigma_{pass1}$ w_i + Σ_{pass2} w_i
- So I can merge both datasets and …
- $b= \Sigma_{pass\ 1\ and\ 2}\ w_i$
- ditto for many backgrounds… (effective for collaborative work)

Multiple backgrounds

- Such plots can be made directly



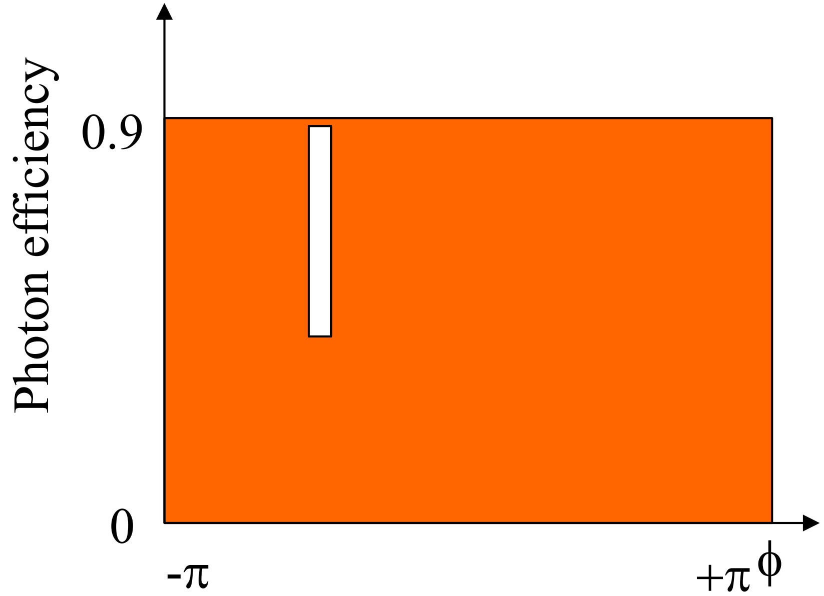
ML Application



- One can increase B1 dataset size and not B2, use weights for proper relative normalisation

Efficiency correction

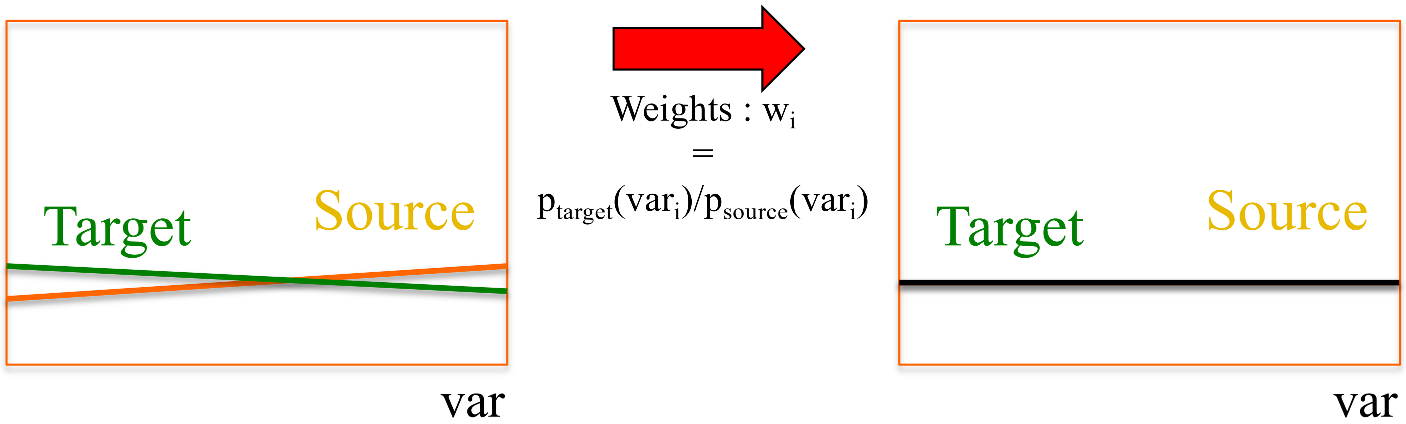
- Suppose, a detector malfunction causes photon efficiency to be halved in a small region of the detector
- ➔resimulate everything taking into account this effect ? (== billion of compute hours)



- No, simply define a new weight:
  - w_i^{photon}=0.5 if one photon in that region, 1 elsewhere
- Then $w_i^* = w_i^{photon}$ 🡐 weights of different source can be multiplied
- And voilà, all event counting, all distributions are automagically corrected
- Particularly handy in large collaborations where many teams work on different aspect of event detection.
  - Each team comes up with its own weight
  - Physicist doing analysis kind (almost) blindly the weights he is given

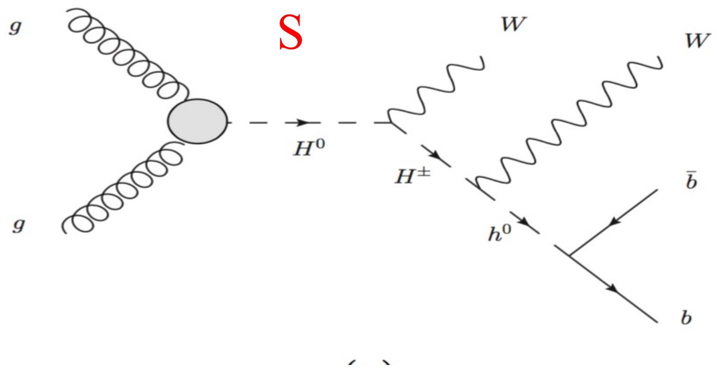
General Re-weighting

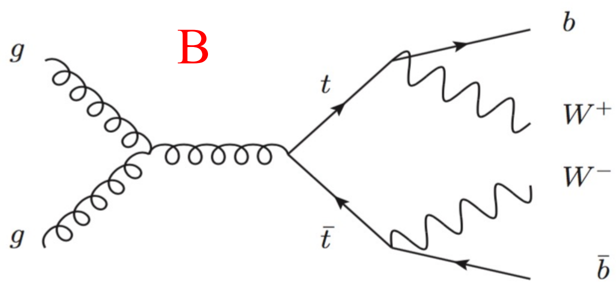
- Suppose a feature distribution is slightly different between a Source (e.g. Monte Carlo) and a Target (e.g. real data)
  - ➔reweight!   …then use reweighted events



Event Generator weight

- Generators are software which creates event with multi particle final states with very precise correlation
- ➔weighted events (weights can even be negative!) (➔more in Andy's talk)





Uncertainty

- When counting unweighted events uncertainty (Poisson case):
  - $N_{pass}=\Sigma_{pass}\ 1$
  - $\sigma\ N_{pass}=\sqrt{N_{pass}}=\sqrt{\Sigma_{pass}\ 1}$
  - $\sigma\ N_{pass}\ /\ N_{pass}=1/\sqrt{N_{pass}}$
- For weighted events (Poisson, binomial more involved):
  - $N_{pass}=\Sigma_{pass}\ w_i$
  - $\sigma\ N_{pass}=\sqrt{\Sigma_{pass}\ w^2_i}$
  - $\sigma\ N_{pass}\ /\ N_{pass}=\sqrt{\Sigma_{pass}\ w^2_i}\ /\Sigma_{pass}\ w_i$
  - Note : if $w_i=1$ ➔ like unweigh power 2!!!
  - Note : if I scale all weights by a : $\sigma\ N_{pass}\ /\ N_{pass}$ is unchanged (as expected)

$$\frac{2+1}{2} = ?$$

$$\frac{2+1}{2} = 1.5$$

$$\frac{\sqrt{2^2 + 1^2}}{2} = ?$$

$$\frac{\sqrt{2^2+1^2}}{2} = 1.118$$

Effective number of events

- Suppose I have 2, and I add 1 (50%) in quadrature? What is the percentage increase ? (5 seconds)
- 12% ! $\sqrt{(2^2+1^2)}/2= \sqrt{5}/2=1.118$
- Meaning : quadratic sum is dominated by the largest values
- ➔having large weights destroy the statistical sensitivity
- Effective number of events of a sample == number of events of an equivalent weightless sample bringing the same precision
  - $N_{eff}= \Sigma^2 w_i /\Sigma w^2_i$
  - $N_{eff}/N=1/(1+Var(x)/<x>^2) <1$
  - The larger the distribution of weights the larger the loss of sensitivity

Caveats

- Reweighting applicable for small-ish corrections (otherwise variance of weight too large➔loss of sensitivity)
- Of course cannot "invent" events
- Not really suitable to rescale variables (if says Energy of a particle is wrong by 2%, better rescale energy directly)
- Also weights are ~easy to compute if uncorrelated
- If correlated, can do 2-dimension reweighting more difficult (curse of dimensionality)
- Beware : not all software tools handle weights correctly, most tools do not handle negative weights correctly

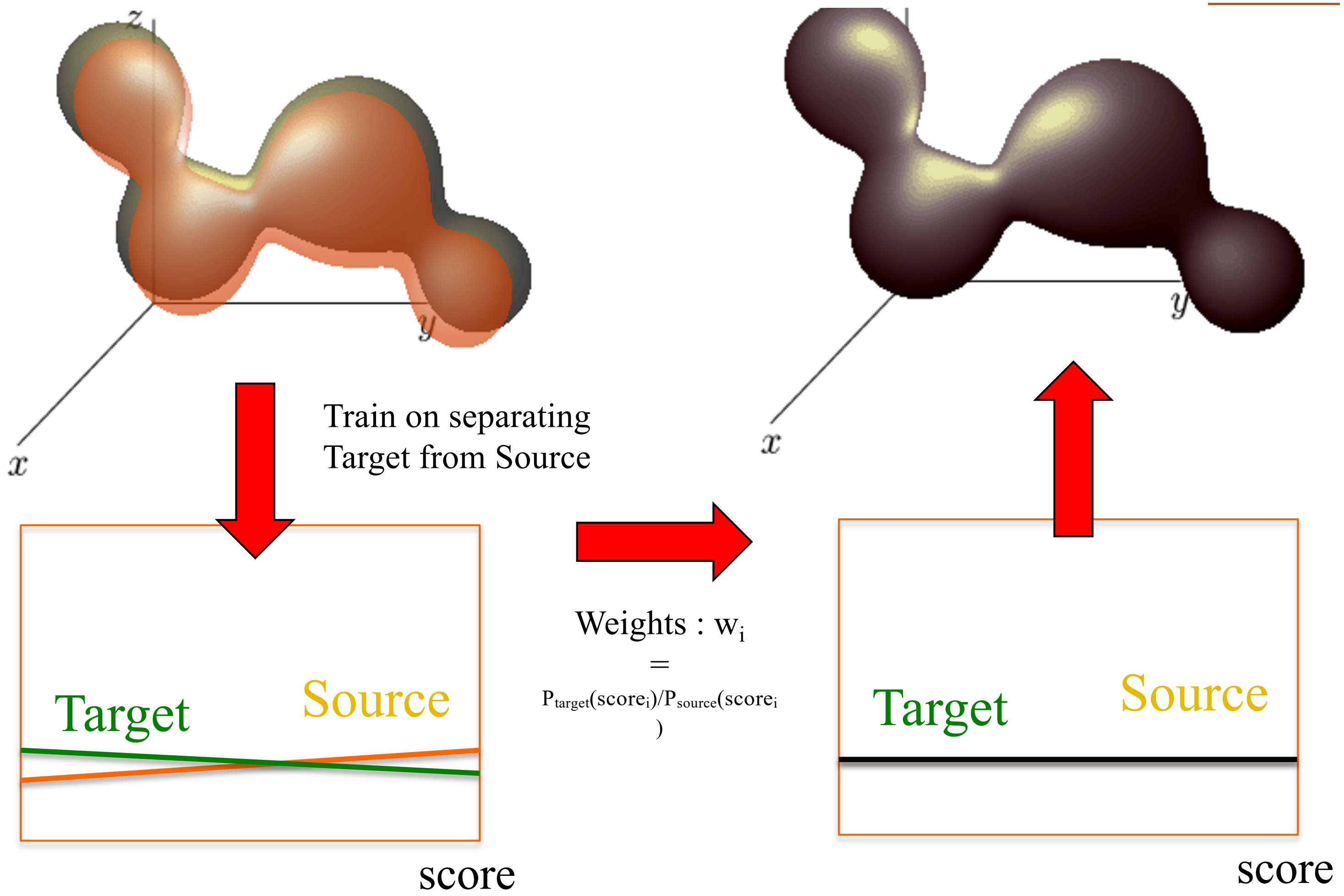
Weights : w_i

$=$

$p_{target}(var_i)/p_{source}(var_i)$

ML for Multidim reweighting

See demo on Andrei Rogozhnikov github and also Kyle Cranmer's github Related : uBoost



ML Multi dim reweighting (2)

- Reweighting the Source distribution on the score allows multidimensional reweighting without statistics problem
- Usual caveat still hold : Target support should be included in Source support, distributions should not be too different otherwise unmanageable very large or very small weights
- (Note : “reweighting” in HEP language <==> “importance sampling” in ML language)
- Only very recently used