An example of background suppression in physics searches

aka

Introduction to statistical analysis in HEP

Collider Physics lectures

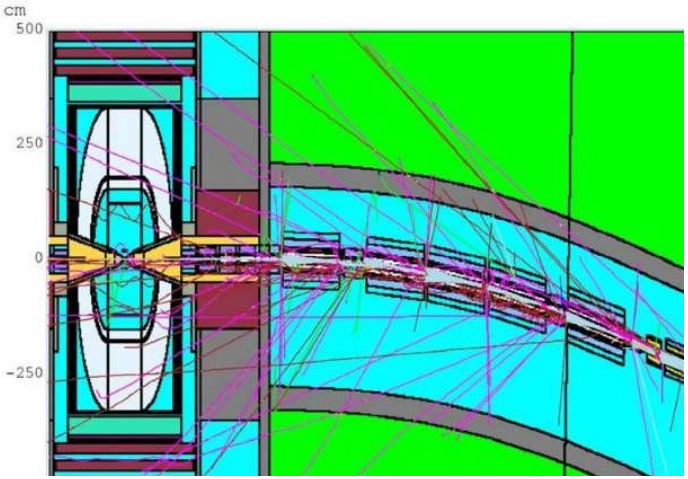
R. Venditti, F. Simone

A.A. 2023-2024

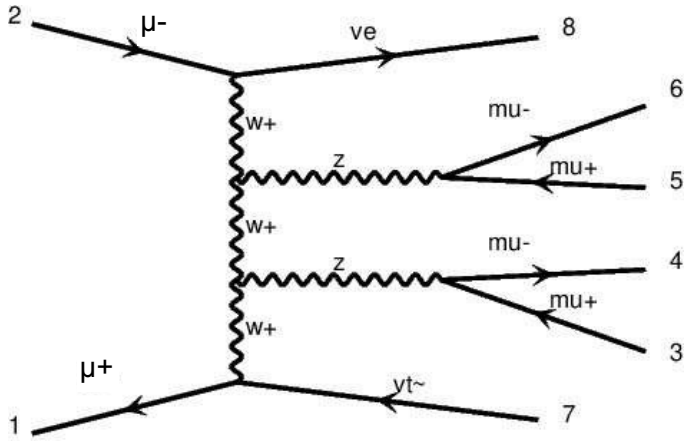
$H \rightarrow ZZ^* \rightarrow 4\mu$ Signal and background

Two main background sources:

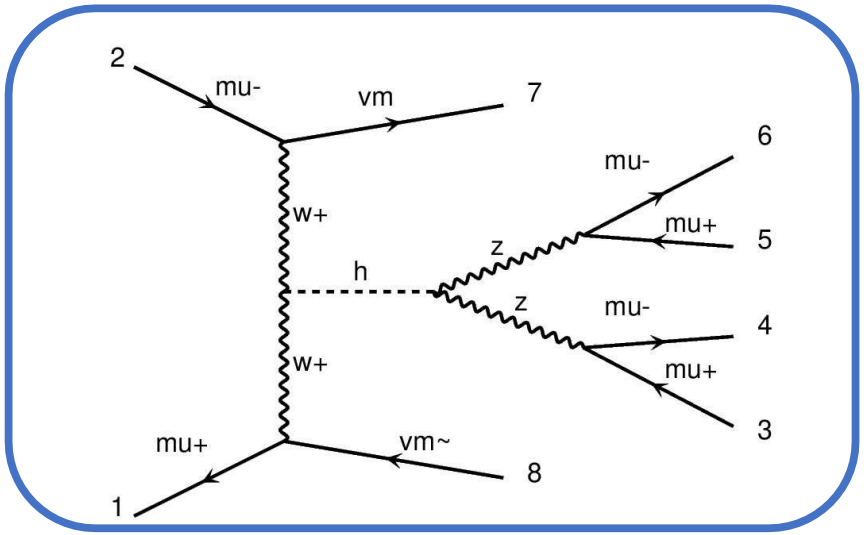
- Beam induced background (BIB):
  - Muon collider: coming from muon beams that decays into electrons, it features the environment of any collision
  - At hadron collider are pile up and Underlying events

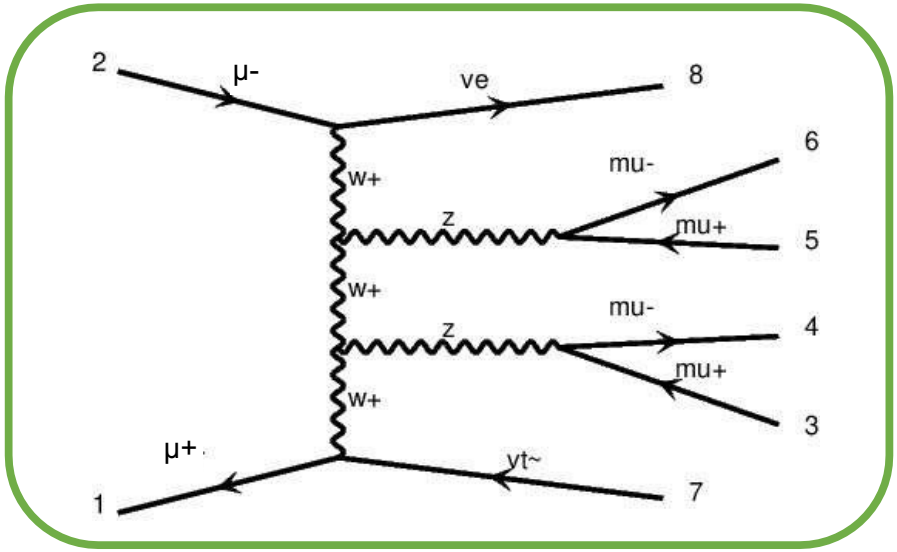


- Standard Model background: physics process from Standard model that have a very similar final state (4 muons) as the signal one.
  - Reducible: can be discriminated by signal
  - Irreducible background: more difficult to disentangle from signal
  - SM bkg events are also affected by the BIB



$H \rightarrow ZZ^* \rightarrow 4\mu$ Signal and SM background





➢ SIGNAL PROCESS:

$\mu^+\mu^- \rightarrow H\nu_\mu\overline{\nu_\mu}$

$\hookrightarrow ZZ^* \rightarrow 4\mu$

➢ SM IRREDUCIBLE BACKGROUND:

➢ Mainly to double Z production

$\mu^+\mu^- \rightarrow 4\mu\ \nu_\mu\overline{\nu_\mu}$

~ 5000 Feynman diagrams

SIGNAL $H \rightarrow ZZ^* \rightarrow 4\mu$

$\sqrt{s}$ (TeV)	L (fb^{-1})	σ (fb)	Expected events
1.5	500	$9.14 \cdot 10^{-3}$	4.57
3.0	1300	$1.47 \cdot 10^{-2}$	19.16

4μ-SM BACKGROUND

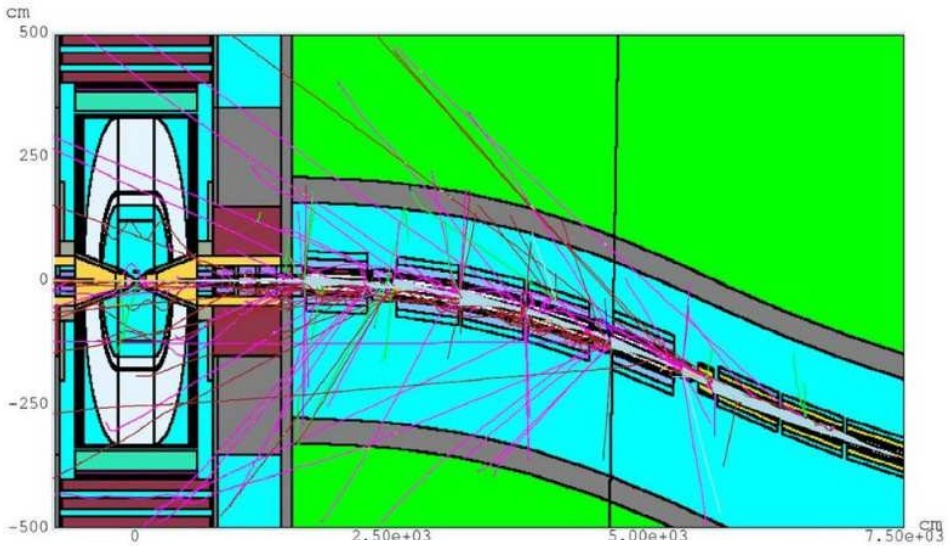
$\sqrt{s}$ (TeV)	L (fb^{-1})	σ (fb)	Expected events
1.5	500	$9.18 \cdot 10^{-3}$	4.59
3.0	1300	$1.79 \cdot 10^{-2}$	23.21

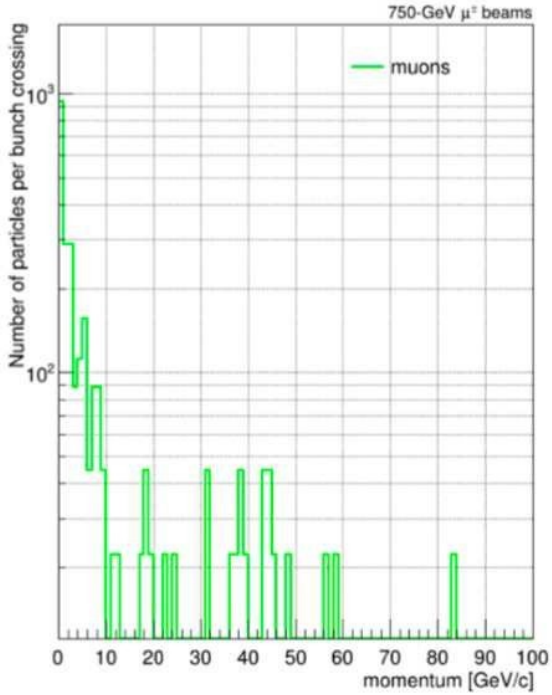
$H \rightarrow ZZ^* \rightarrow 4\mu$ Signal and SM background

- μμ→ZZ→4μ events are produced with same rate as signal
- Signal and background have
  - same order of magnitude
  - same final states (4 muons)
- We cannot get rid of the SM background→ in real data signal and SM background events are mixed in agreement to their cross section
  - → need to find powerful methods to reject the SM background events while keeping signal events
- Two main methods
  1. Apply topological selections in AND:
     - study the distributions of kinematic and event variables of signal and background in order to find those that are able to discriminate between the two processes
     - Apply selections on variable at time
  2. Combine all the variables in a single discriminator used to apply selection or extract signal

The beam induced background

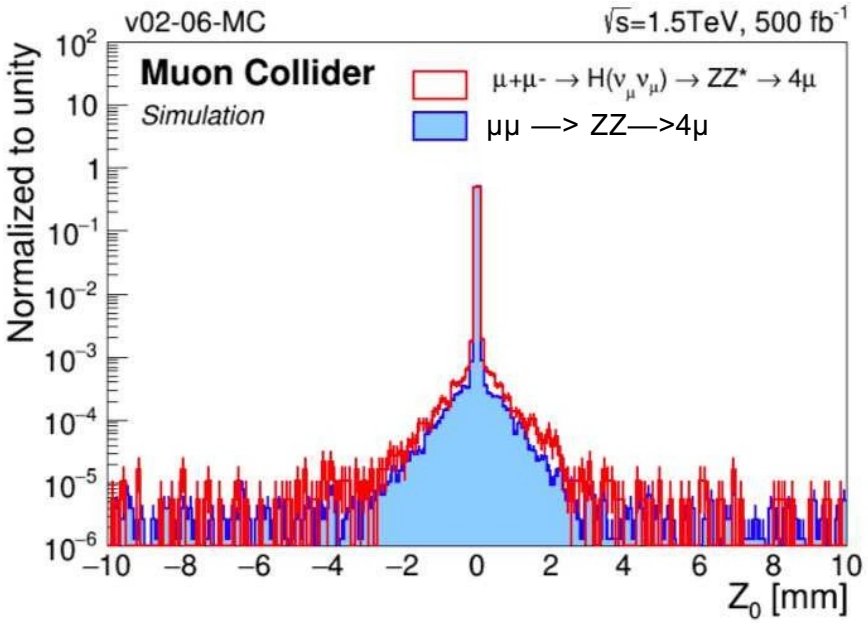
- Due to the muon decay $\mu \rightarrow e\nu_\mu\nu_e$.
- Electrons from muon decays have a mean energy of approximately 1/3 of that of muons,
  - 0.75-TeV muon beams→250 GeV.
  - These electrons travel towards the beam pipe/detector material and emit energetic synchrotron photons tangent to the electron trajectory.
  - Electrons and photons initiate electromagnetic showers that result in intense fluxes of muons, hadrons and daughter electrons and photons.
    - The spectrum of Muons produced from electro/photon showering lies in the GeV range (same as H--> ZZ--> 4μ signal)
  - → these particles are produced during each collision event, therefore are super-imposed to the hard scattering process!!!



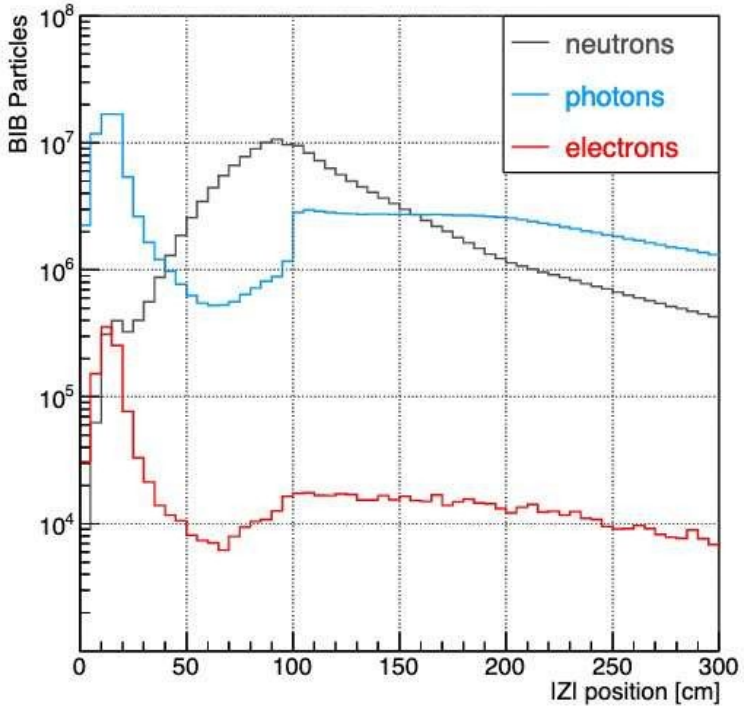


$H \rightarrow ZZ^* \rightarrow 4\mu$: BIB suppression

- In order to reduce the background due to muons from BIB, we simulated and studied the kinematic distributions of
  - signal without BIB
  - signal + BIB



Longitudinal (along z axis)distance of final state muons to beams impact point (signal and μμ→4μ bkg)



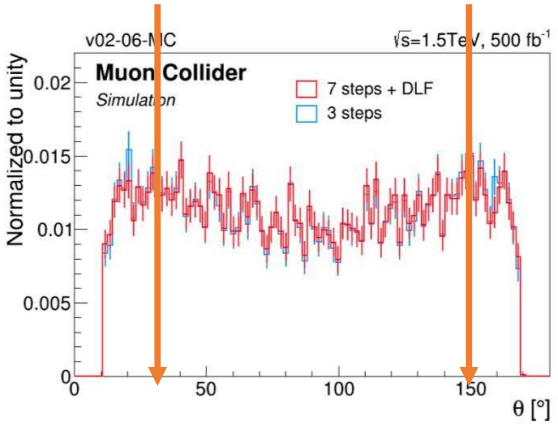
Longitudinal distance of BIB particles to beam impact point

Same in transversal (x-y) plan!!!

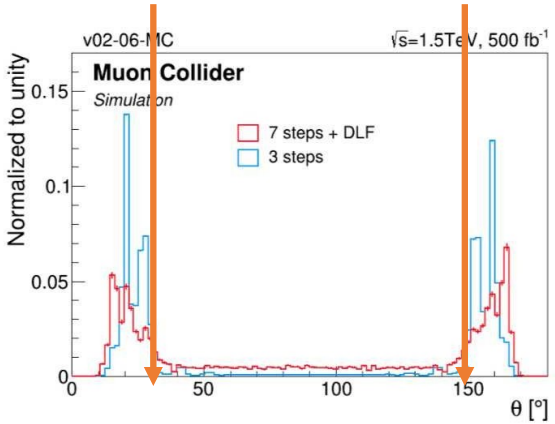
$H \to ZZ^* \to 4\mu$: BIB suppression

Good quality final state muons:

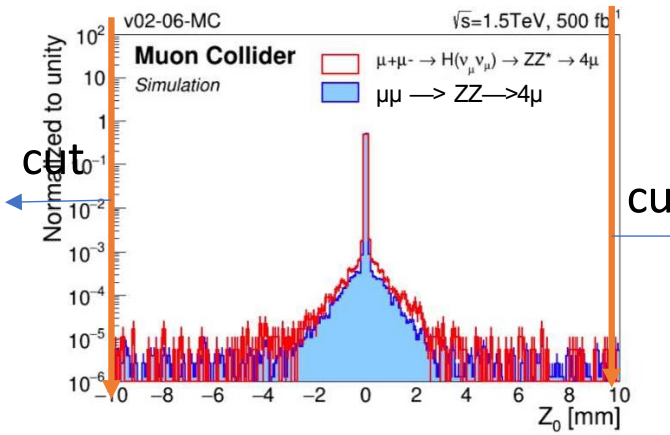
- $p_T > 5\ GeV;\ |\eta| < 2.5;$
- $D_0 < 2mm;\ Z_0 < 10mm;$

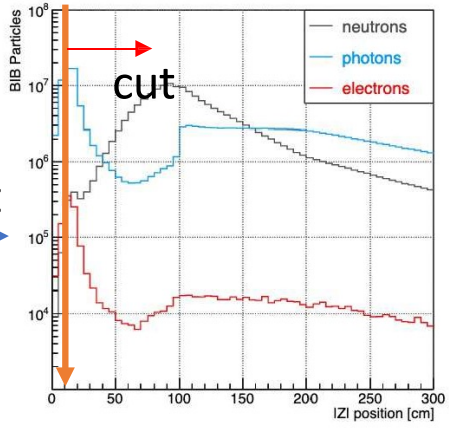


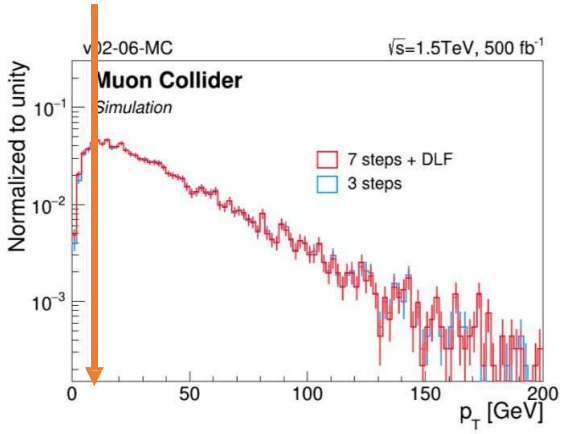
(a) only sig



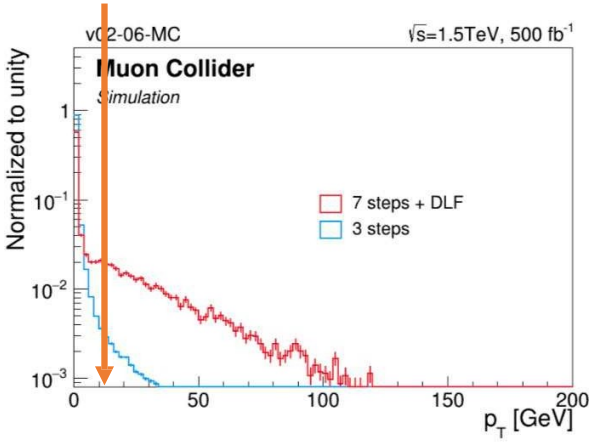
(b) sig+BIB







(a) only sig

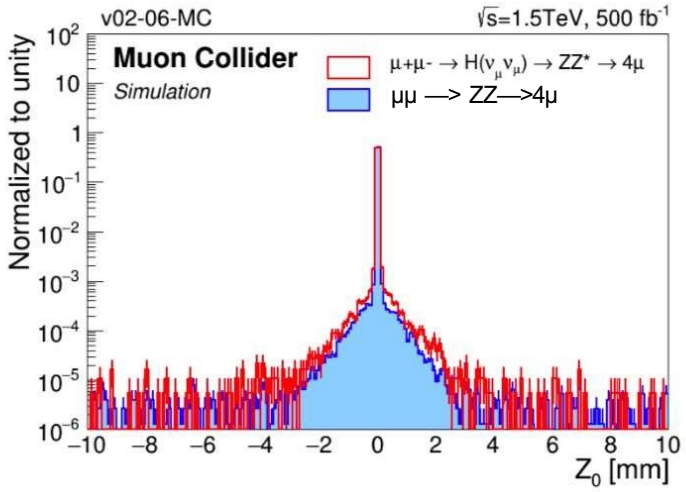


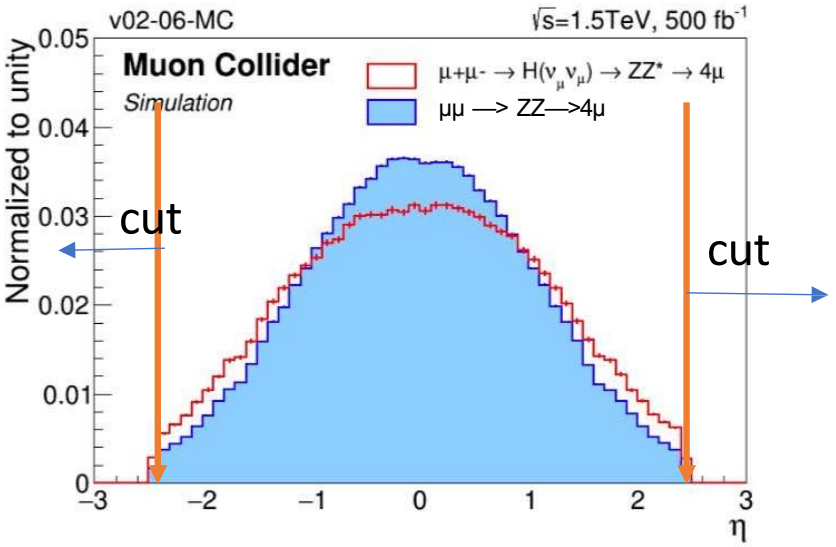
(b) sig+BIB

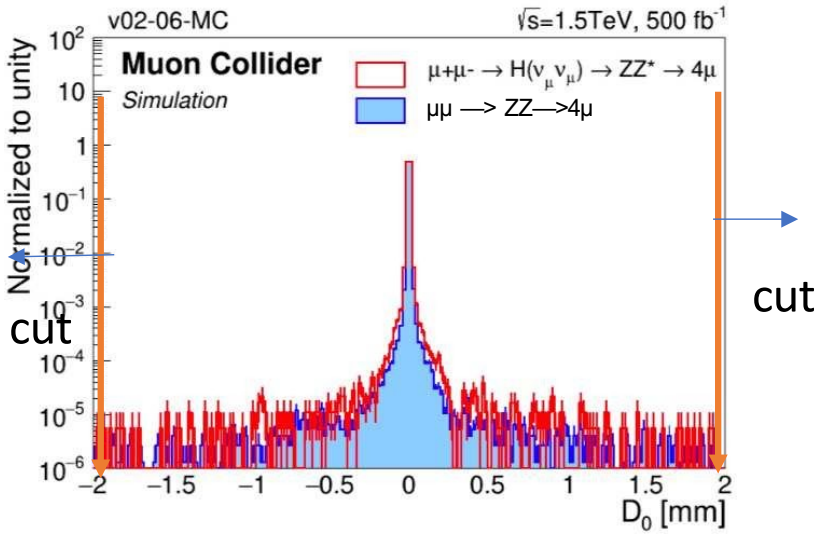
Effect on irreducible bkg

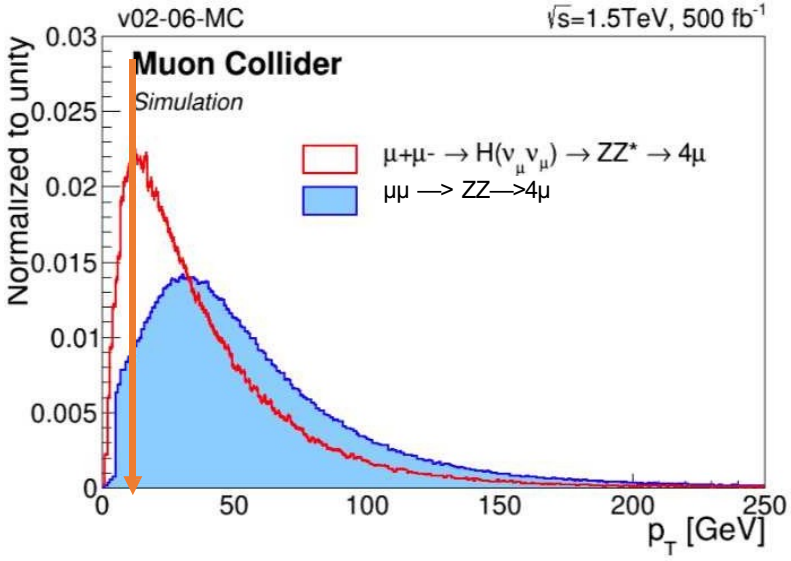
Good quality final state muons:

- $p_T > 5\ GeV;\ |\eta| < 2.5;$
- $D_0 < 2mm;\ Z_0 < 10mm;$





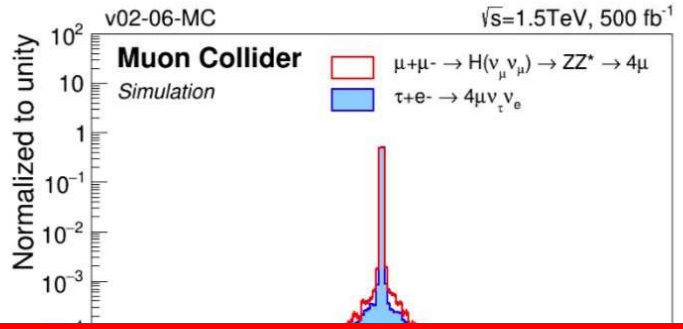


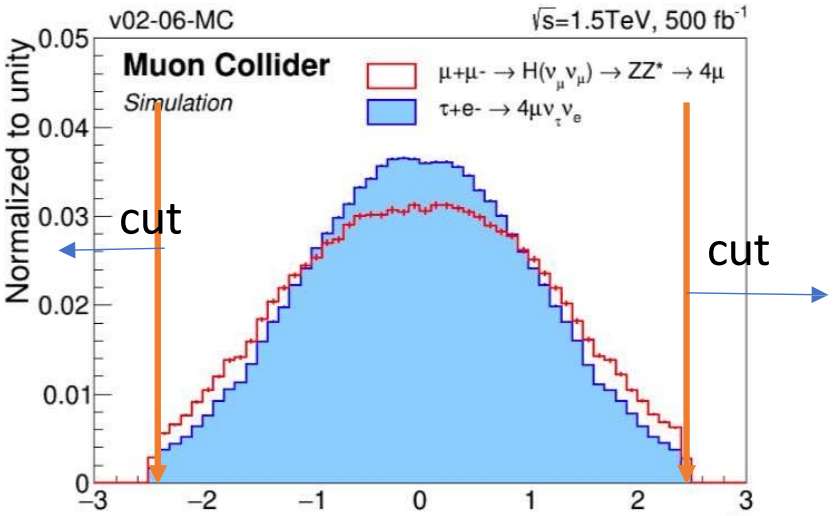


Effect on irreducible bkg

Good quality final state muons:

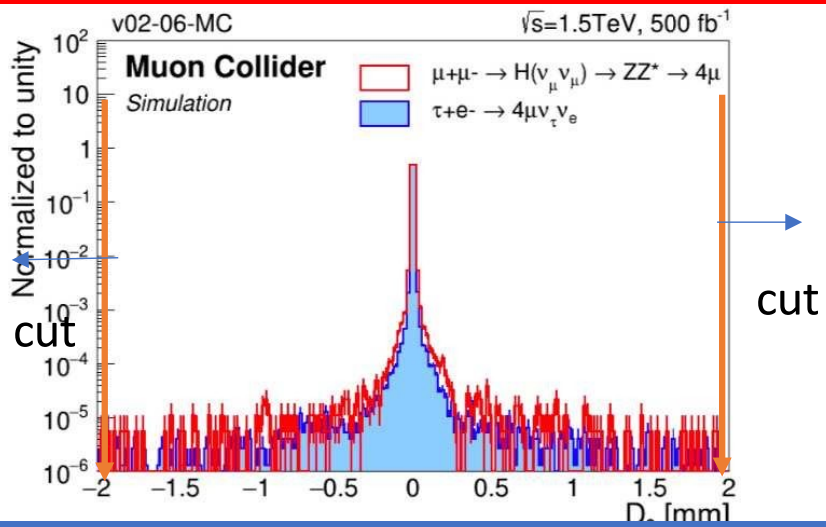
- $p_T > 5\ GeV;\ |\eta| < 2.5;$
- $D_0 < 2mm;\ Z_0 < 10mm;$

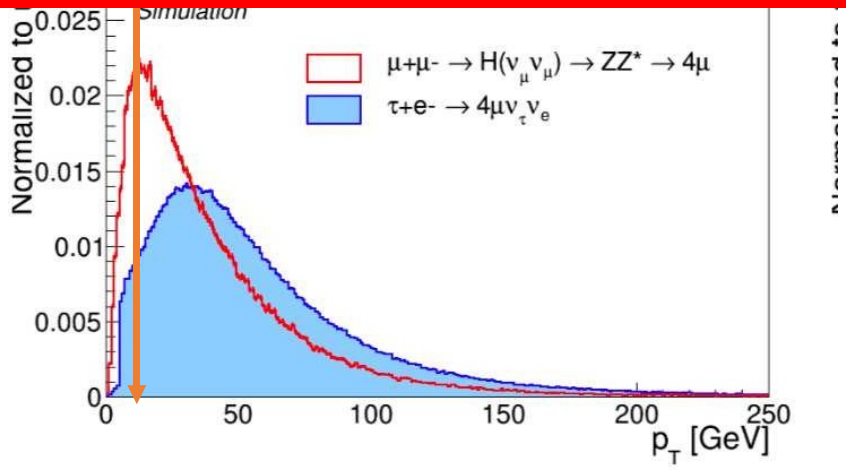




No effect on irreducible bkg

→ (this is why it is called irreducible!)





$H \to ZZ^* \to 4\mu$ topological selection

Good quality final state muons:

- $p_T > 5\ GeV;\ |\eta| < 2.5;$
- $D_0 < 2mm;\ Z_0 < 10mm;$

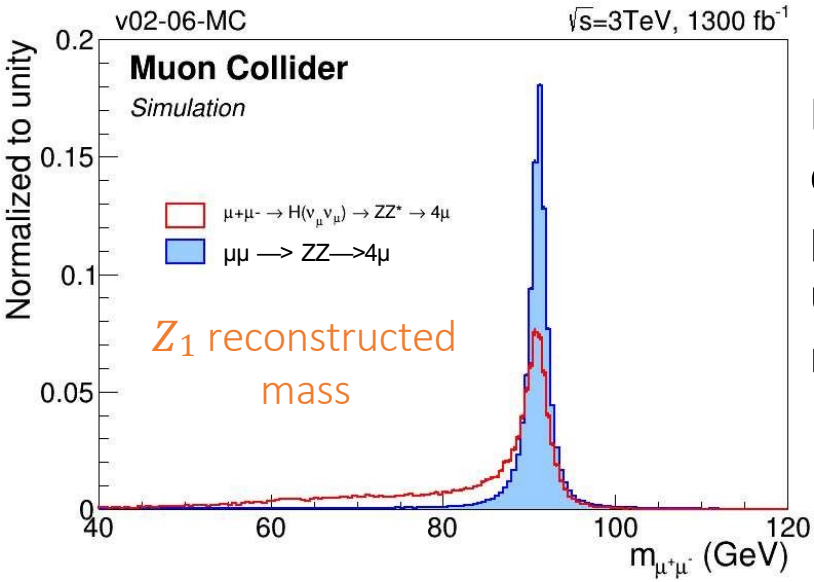
Let's study the distribution of the invariant mass of the Z bosons
- First reconstruct the Z boson starting from muons

Z candidates:

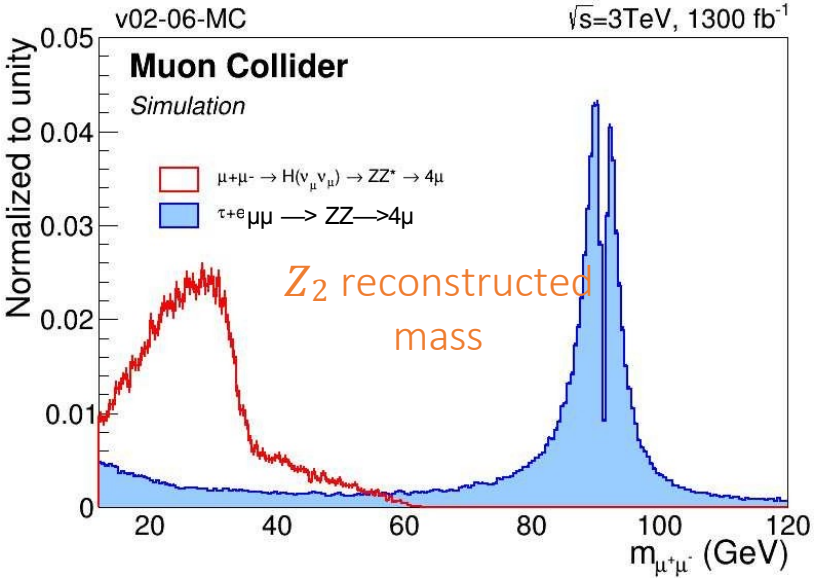
Select OS muon pairs $with\ \ 12 < m_{\mu\mu} < 120\ GeV.$

ZZ candidates: non-overlapping *Z*s

- Z_1: *Z* candidate with mass closest to the nominal value
- Z_2: other *Z* candidate



No discriminating power, just useful to reject muons from BIB



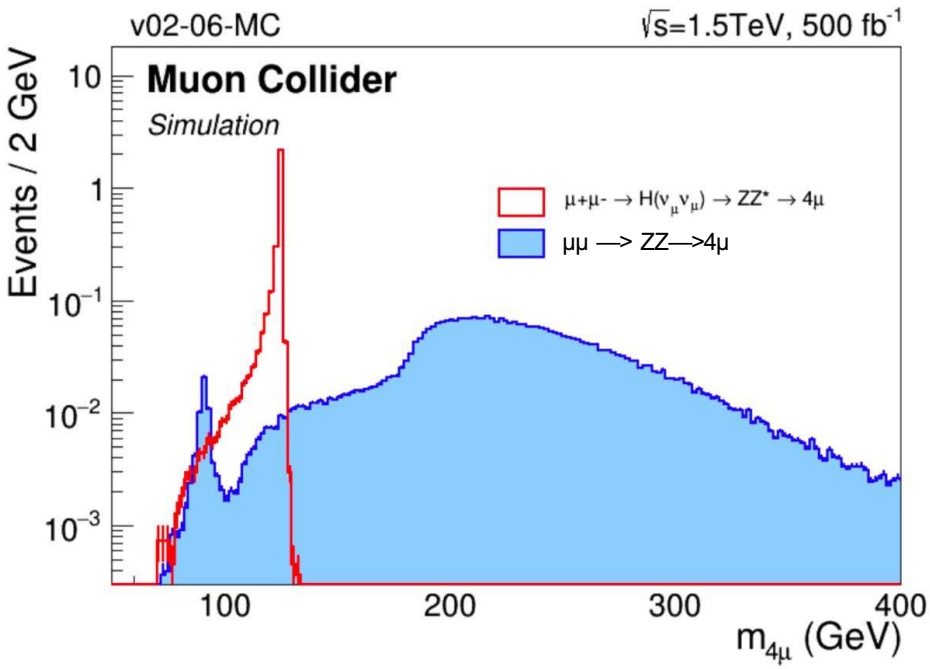
Some discriminating power, but no possible to cut without biasing the 4μ invariant mass (Higgs mass)

$H \to ZZ^* \to 4\mu$ topological selection

Good quality final state muons:
- $p_T > 5\ GeV;\ |\eta| < 2.5;$
- $D_0 < 2mm; Z_0 < 10mm;$

Z candidates:
OS muon pairs $12 < m_{\mu\mu} < 120\ GeV$.
ZZ candidates: non-overlapping *Z*s
- Z_1: *Z* candidate with mass closest to the nominal value
- Z_2: other *Z* candidate

Selection of ZZ candidates:
- $\Delta R(\mu_i, \mu_j) > 0.02$
- $p_{T,\mu_i} > 20$ GeV
- $p_{T,\mu_j} > 10$ GeV
- Z_1 mass > 40 GeV
- $m_{4\mu} > 70$ GeV
- Arbitration based on Z1 mass



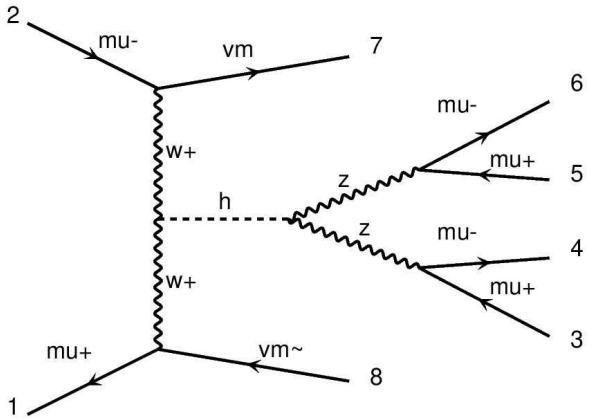
$\sqrt{s} = 1.5$ TeV

Signal	Background
Events	Events
2.97±0.02	3.93±0.01

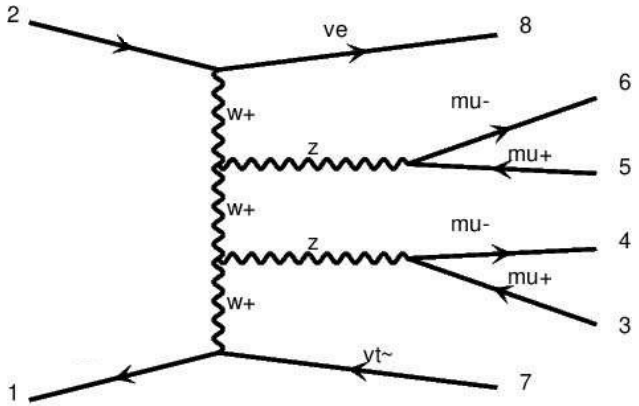
Signal and bkg have still comparable yields!

Finding difference in signal and bkg

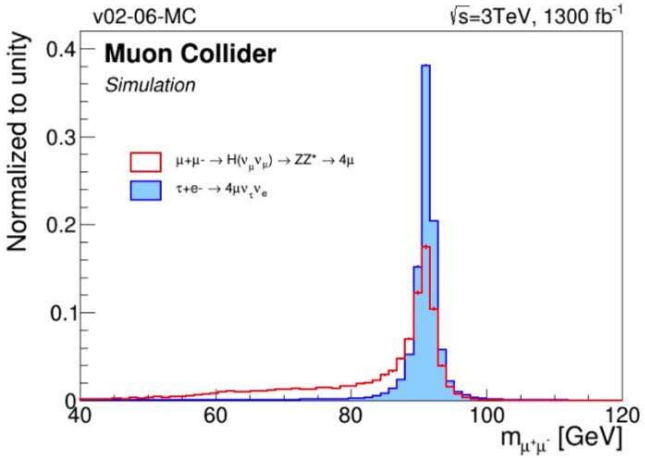
- Signal and background final state muons cannot be discriminated on the basis of simple kinematic variables, like pT and η.
- The first differences between the two classes of events emerge from the **invariant mass distributions of the constructed muon combinations.**
- Other difference come from the physics that rules the two processes



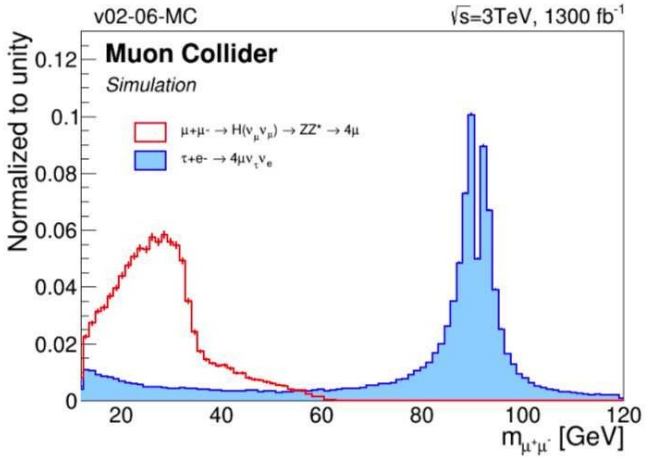
- **signal** channel H → ZZ* → 4μ: when we have a resonance X decaying in two gauge bosons which subsequently decay in four leptons, the angular distributions of the decay products contain information about couplings, spins and other quantum numbers of their parent → the angular distributions of the gauge bosons, are expected to be compatible with a 0+ resonance
- for the **background** channel ZZ → 4μ, the gauge bosons are not produced from the decay of a resonance, and we expect different distributions



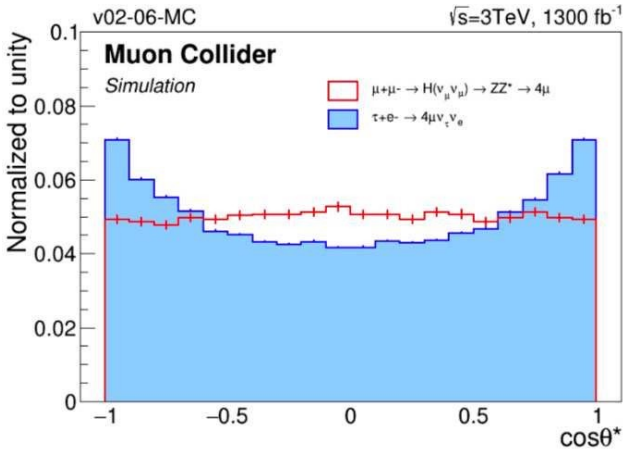
Finding difference in signal and bkg

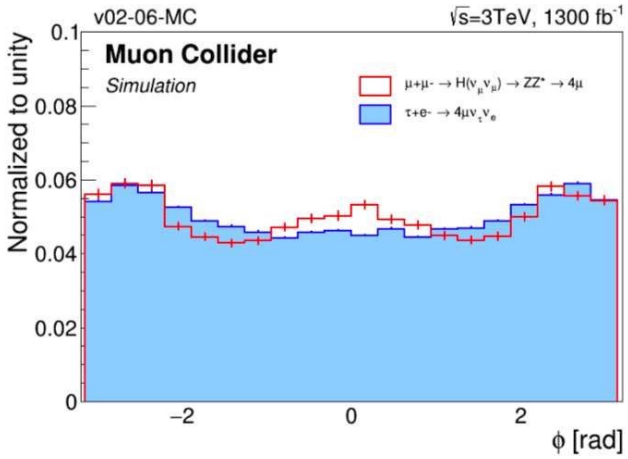


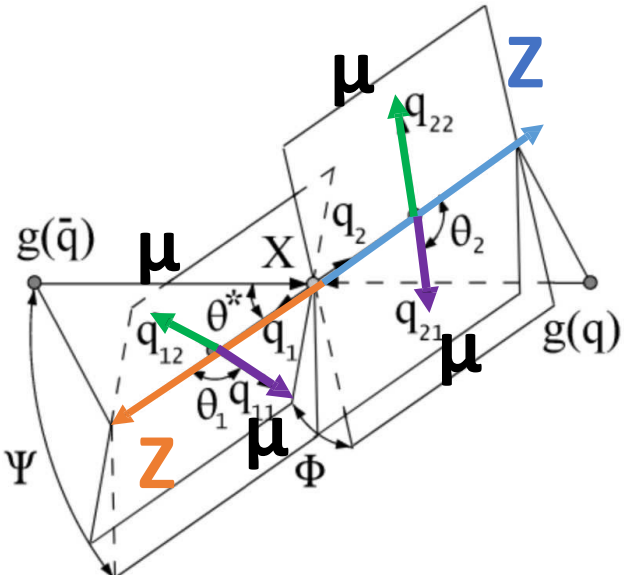
(a) Z_1 mass



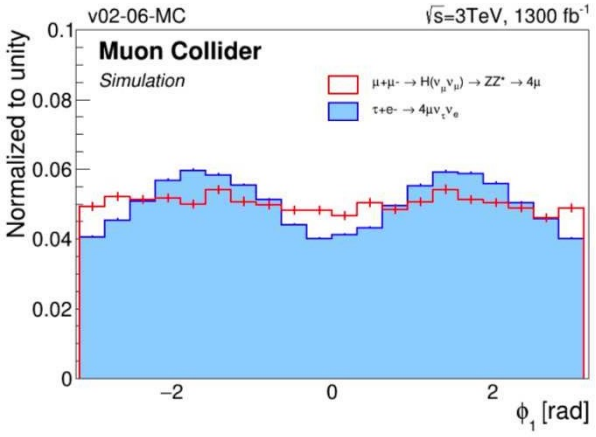
(b) Z_2 mass



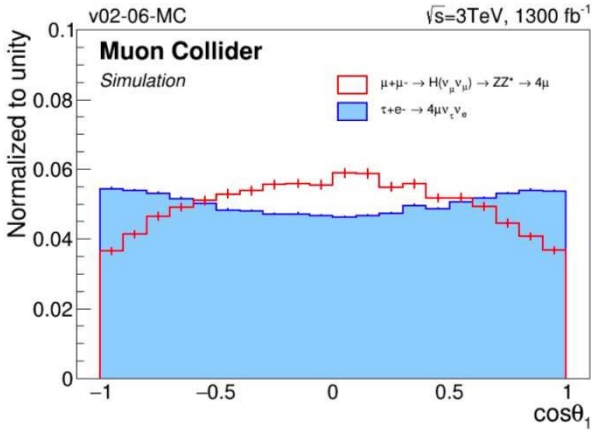




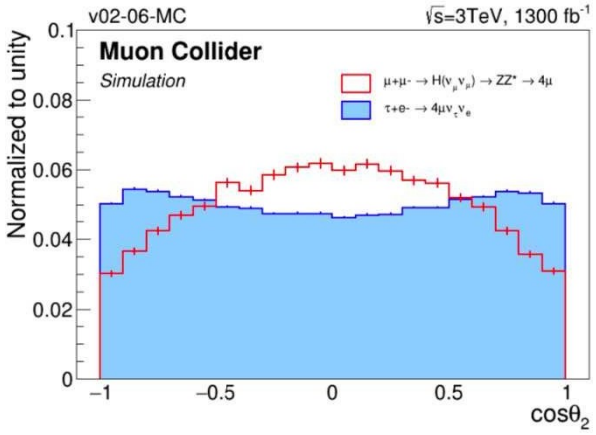
Finding difference in signal and bkg



(e) Φ_1

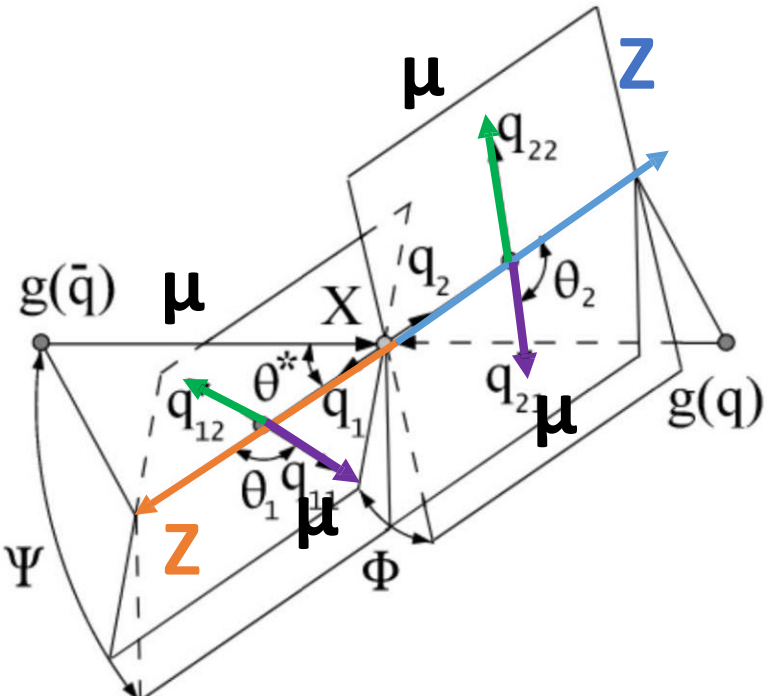


(f) $\cos\theta_1$

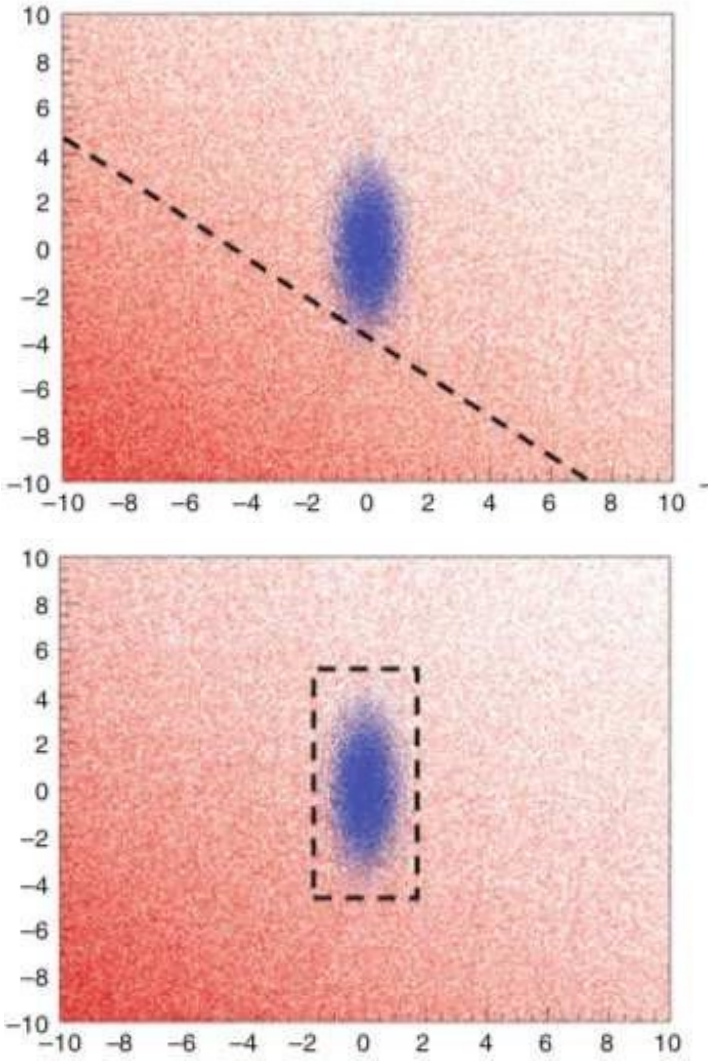


(g) $\cos\theta_2$

- Domain of the distribution of these variables is quite similar for signal and bkg → a selection is not super-useful
- Instead the shape of the distribution is different →value of these observables is different between sign and bkg on event-by-event basis



Multivariate analysis

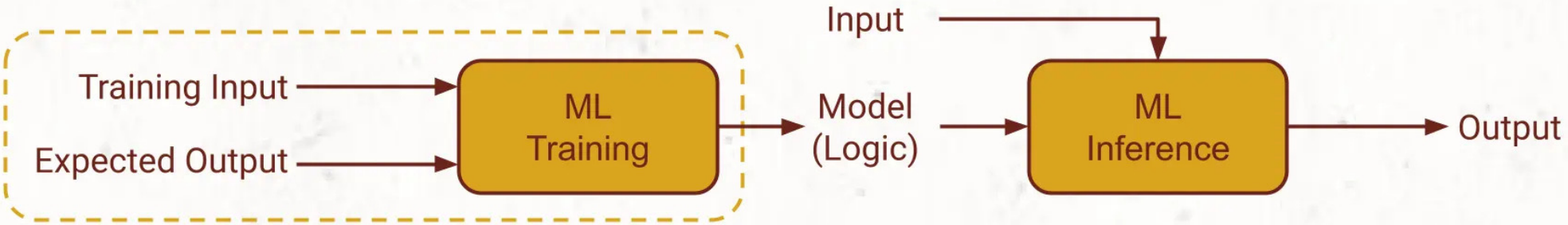


- Often there is not just a single variable on which we have to define a threshold for separating signal and bkg
  Possible approaches:
  - Define multidimensional selections (as in the picture) → optmization is difficult
  - Combine all variables (features) in one!!!!
- The problem of classifying data according to the value of multiple variables is called *multivariate analysis* (MVA)
- MVA is implemented by mean of computer algorithms tha receives as input a set of discriminating variables
  - Each variable ALONE does not allow to reach an optimal selection power.
  - The algorithm computes an output that combines the input features.
- The discriminant output (discriminator, score or classifier) has high discriminating power and is used to perform the signal selection
- The choice of the input variables is a key task of the algorithm since an optimal choice allows achieving the best possible performances.

Machine learning

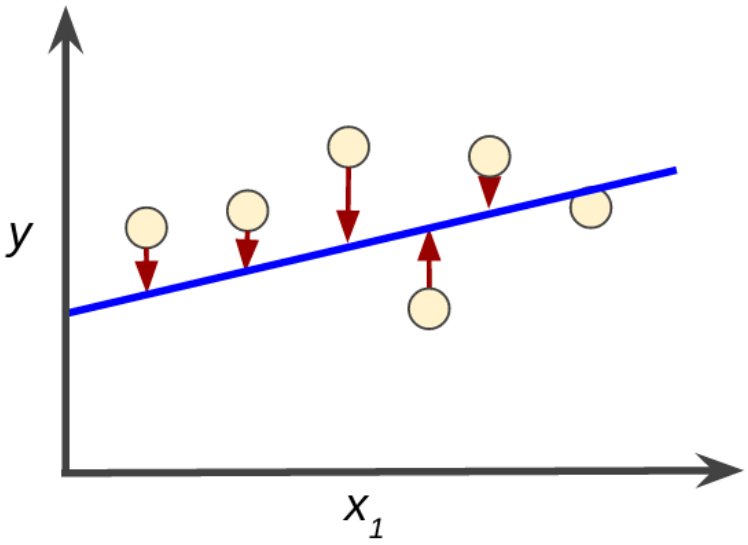
Traditional Programs: Define algo/logic to compute output

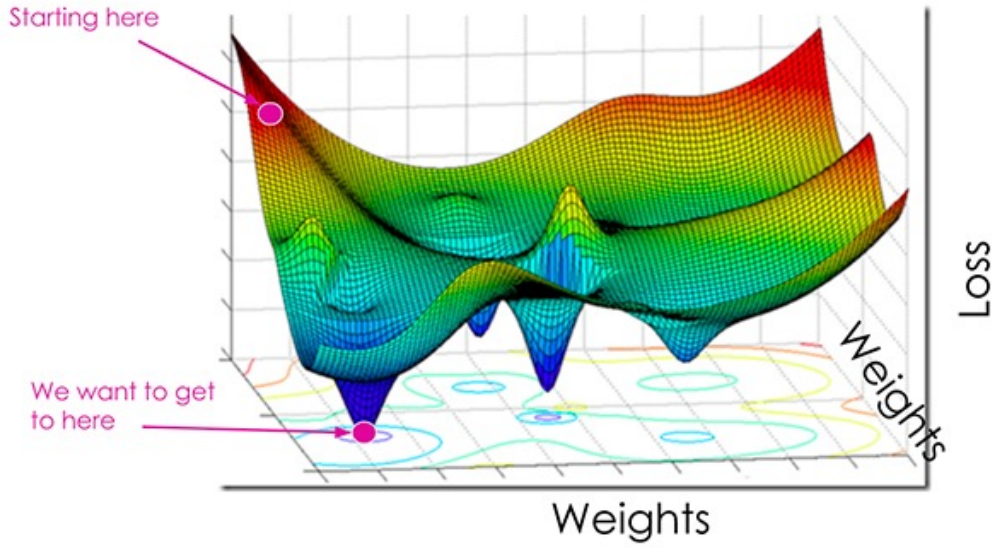
Machine Learning: Learn model/logic from data



Procedure

- Model: mathematical assumption to describe our data (e.g. for regression, a linear function)
- Model parameters (or weights): a set of scalars defining the model, $\theta = \{\theta_1 \dots \theta_N\}$
- Loss function: describes the «distance» between labels ($\boldsymbol{y}$) and predicted values ($\hat{y}$). It is a scalar function of θ!
- Training rule and optimisation: loss is minimised iteratively (i.e. in most cases we need to compute derivatives!). At each step, $\{\theta_1 \dots \theta_N\}$ are updated according to some rule. Minimum loss corresponds to optimal set of parameters → we learn the model from data!



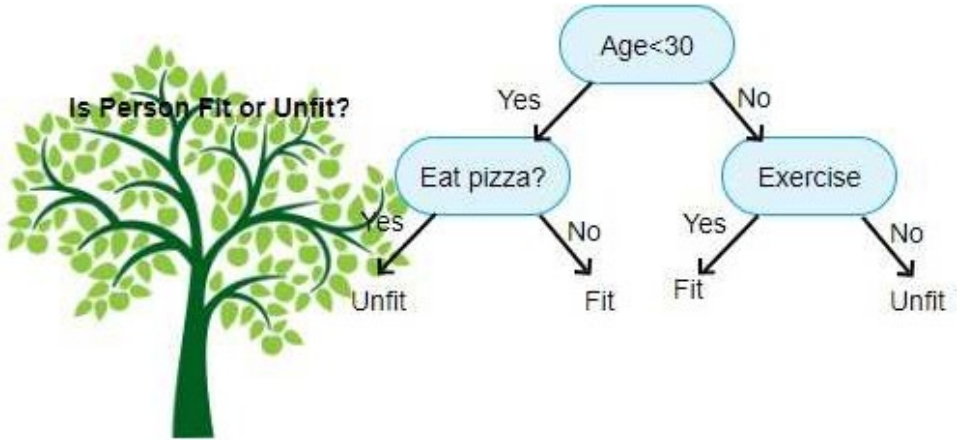


Procedure

- Features ingeneering: Providing as input to the algorithm two large datasets,
  - one distributed according to the H0 (signal) hypothesis
  - one distributed according to H1 (bkg) hypothesis.
- Training (learning): a discriminator is built by using all the input variables.
  - supervised machine learning: the parameters are iteratively modified by comparing the discriminant output to the true origin of the dataset
  - unsupervised machine learning: algorithm group and find patterns in the the data according to the observed distribution of the input data are called
- Test: once a the training has been performed, the output discriminator is computed in a separated, independent dataset for both H0 and H1
- A check is performed between test and training classifier and their performances (in terms of ROC curve are measured)
  - If the test fails and performance of test and training are different, it is symptom of overtraining: our model learning the exact pattern of training examples and will have poor generalisation power on new data

Decision Trees and Forest - Intro

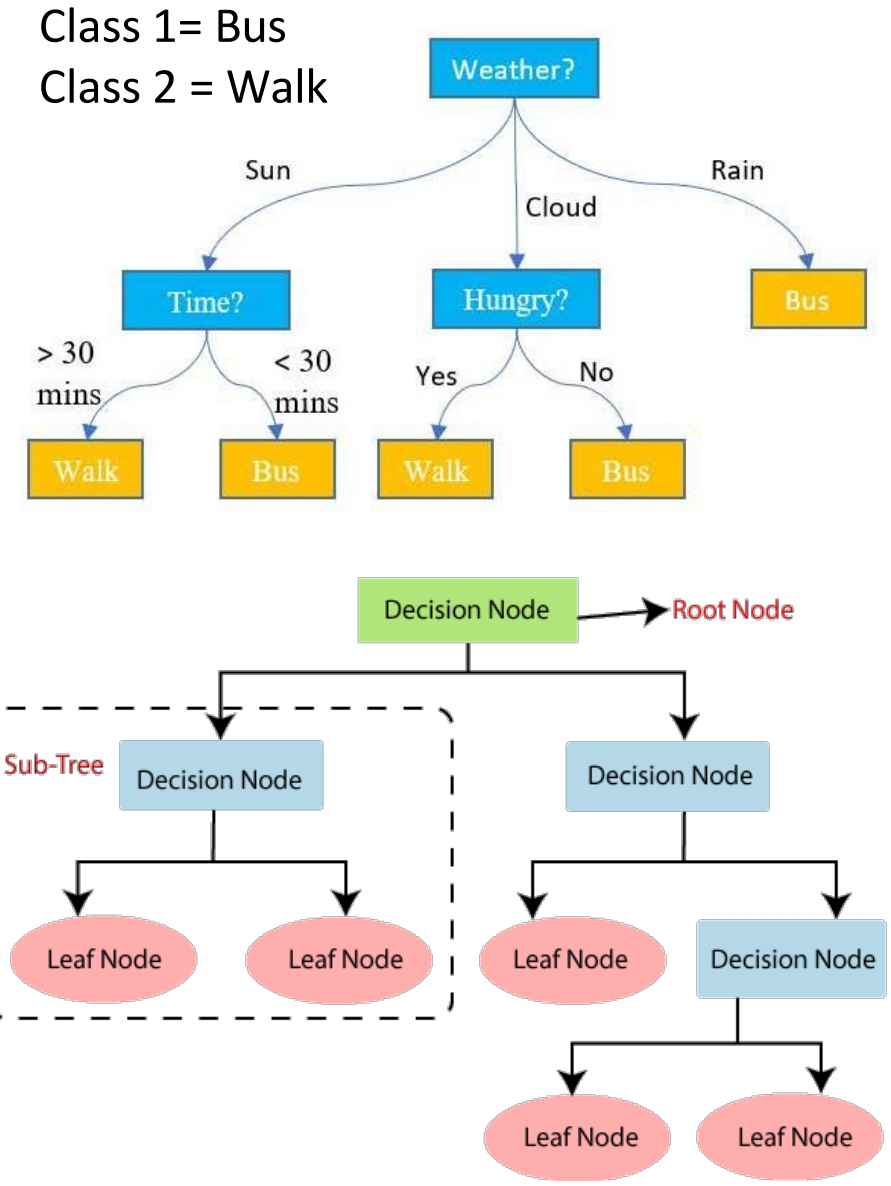
- One of the most popular MVA algorithms is Boosted decision tree
- Decision Trees and their extension Random Forests are robust and easy-to-interpret supervised *machine learning algorithms for Classification*.



- (Deep) Neural Networks are another very common choice for classification problems.
- However, despite their power against larger and more complex datasets, they are extremely hard to interpret and neural nets can take many iterations and hyperparameter adjustments before a good result is had.
- one of the biggest advantages of using Decision Trees and Random Forests is the ease in which we can see what features or variables contribute to the classification and their relative importance based on their location depthwise in the tree.

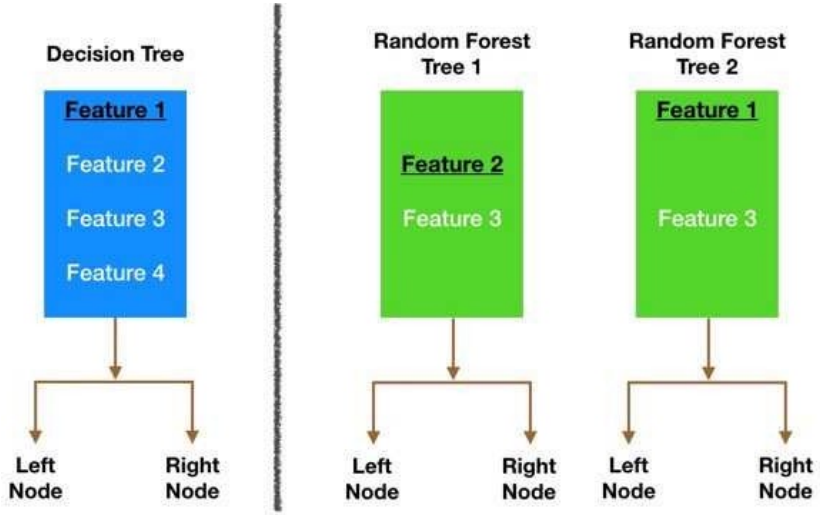
Decision Tree

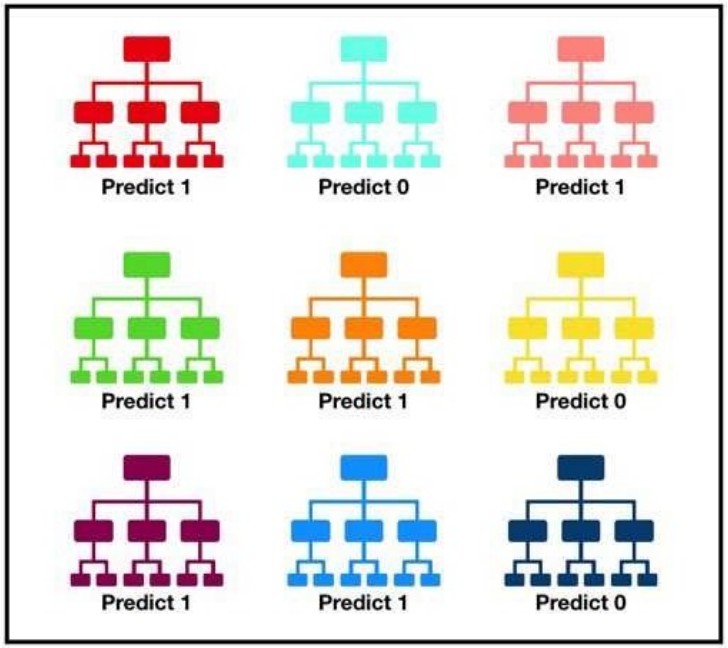
- A *decision tree* is a sequence of selection cuts that are applied in a specified order on a given set of features for both signal and bkg datasets .
- Each cut splits the sample into *nodes*, each of which corresponds to a given number of observations classified as class1 (signal) or as class2 (background).
- A node may be further split by the application of the subsequent cut in the tree.
- Nodes in which either signal or background is largely dominant are classified as *leafs*, and no further selection is applied.
- A node may also be classified as leaf, and the selection path is stopped, in case too few observations per node remain, or in case the total number of identified nodes is too large



From tree to forest

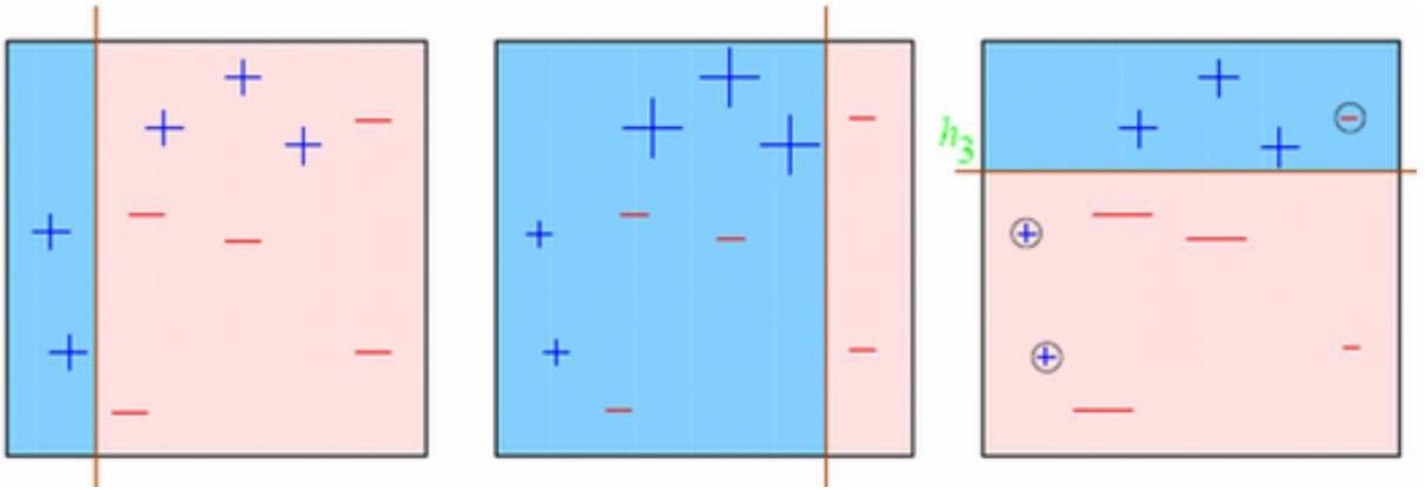
- The random forest algorithm consists of ‘growing’ many decision trees from replicas of the training samples obtained by randomly resampling the input data.
- The final score of the algorithm is given by an unweighted average of the prediction (zero or one) by each individual tree.
- Random forest, consists of a large number of individual decision trees that operate as an ensemble.
- In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the best separation
- In contrast, each tree in a random forest can pick only from a random subset of features.
- that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction).





Boosting

- Boosting is a method of combining many weak learners (trees) into a strong classifier.
- The *boosting* procedure iteratively adds a new tree to a forest obtained by performing the selection optimization after test observations have been reweighted according to the score given by the classifier in the previous iteration.



Box 1 (From the left): Output of First Weak Learner

1. Initially all points have same weight (denoted by their size).
2. The decision boundary predicts 2+ and 5- points correctly
3. Three + are incorrenctly assigned and the algo assign them the highest weight

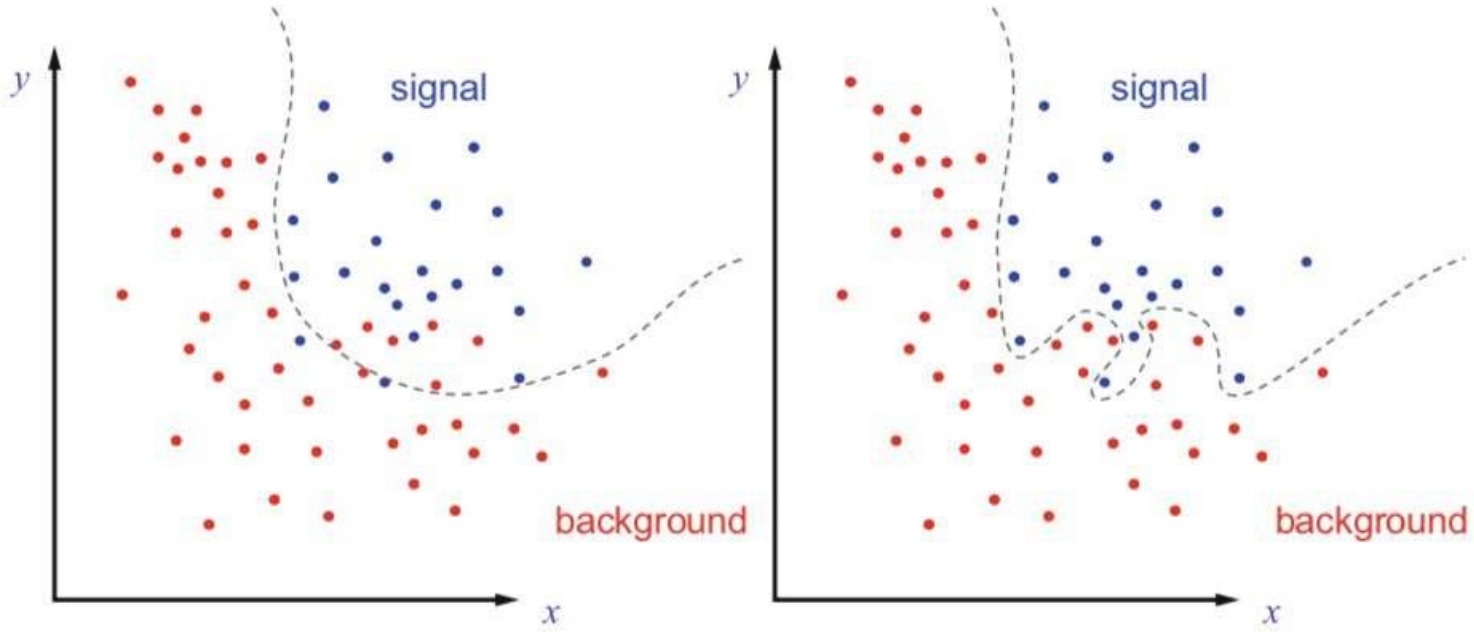
Box 2: Output of Second Weak Learner

1. The points classified correctly in box 1 are given a lower weight and vice versa.
2. The model focuses on high weight points now and classifies them correctly. But, others are misclassified now.

- All models are given a weight depending on their accuracy→consolidated result is generated.

Overtraining

- A potential general problem with the training of machine learning algorithms is the possibility that the algorithm tries to exploit artifacts of the necessarily finite size of training samples that are not representative of the actually expected distributions.
- This possibility may occur, in particular, if the size of the training sample given as input is not very large.
- In practice, an irregular selection may pick or skip individual observations in training categories.



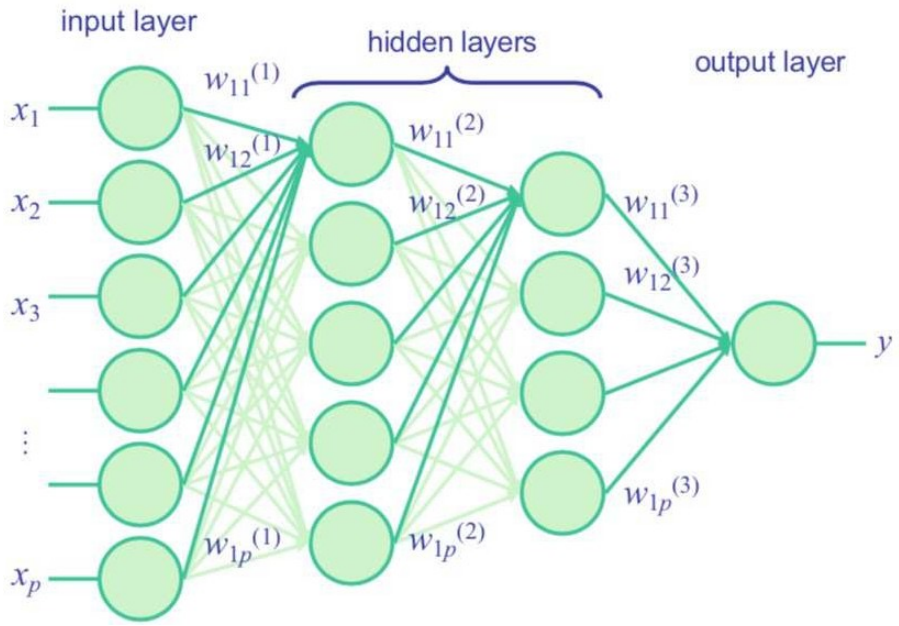
Overtraining

- The algorithm needs thousand of examples to learn and it is crucial that the training data be representative of the new cases we want to generalize to.
- If the sample is too small, we will have sampling noise (i.e., nonrepresentative data as a result of chance), but even very large samples can be nonrepresentative if the sampling method is awed. This is called sampling bias.

- The only way to know how well a model will generalize to new cases is to actually try it out on new cases.
- Common practice: split the data into two datasets,
  - Use the training set to train model
  - Use the test set to test it.
- Disagreement in performance of the classifier are a hint of possibile overtraiining
- Performance to be checked:
  - Agreement between classifier distributions in test and training
  - ROC curve agreement in test and training
- Hence of overtraining is spotted by looking at the discriminant distribution in the training sample and compare with another independent *test sample*

Neural Networks

- Artificial neural networks are one of the most popular
  - Based on the neuron concept: take N features as an input, each neuron combine them into an output in a recursive way
  - Learning: The combination is continuously updated by minimizing the difference between the **NN output F(x) and the true value of the classifiere** (loss function)

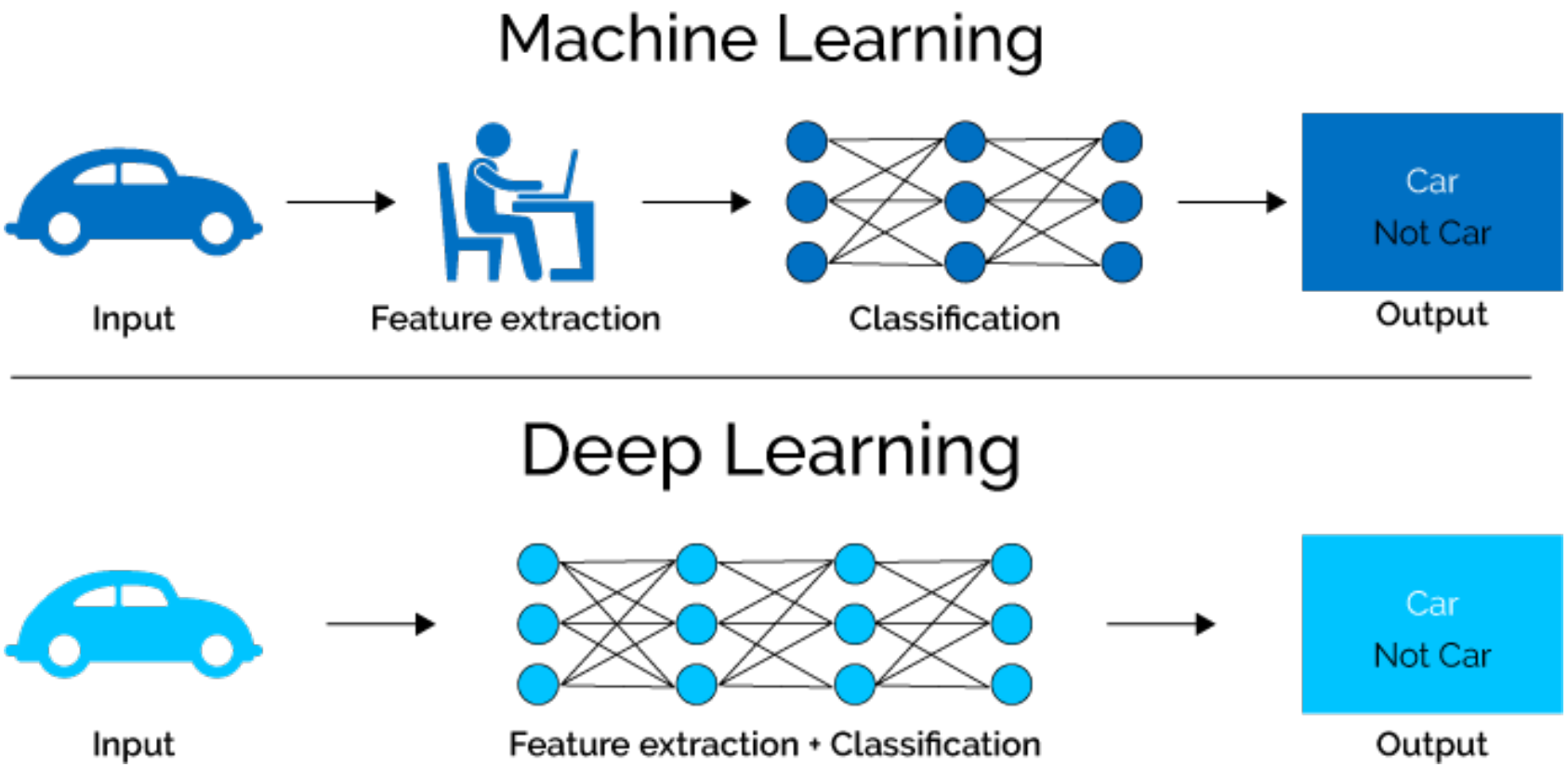


- Input variable values
  *x*1; x2 ;…; *xp*

are passed to a first *input layer*, whose output is passed as input to the next layer, and so on.

- The last *output layer* is usually constituted by a single node that provides the discriminant output.
- Intermediate layers between the input and the output layers are called *hidden layers*.
- Such a structure is also called *feedforward multilayer perceptron*.

Machine and deep learning



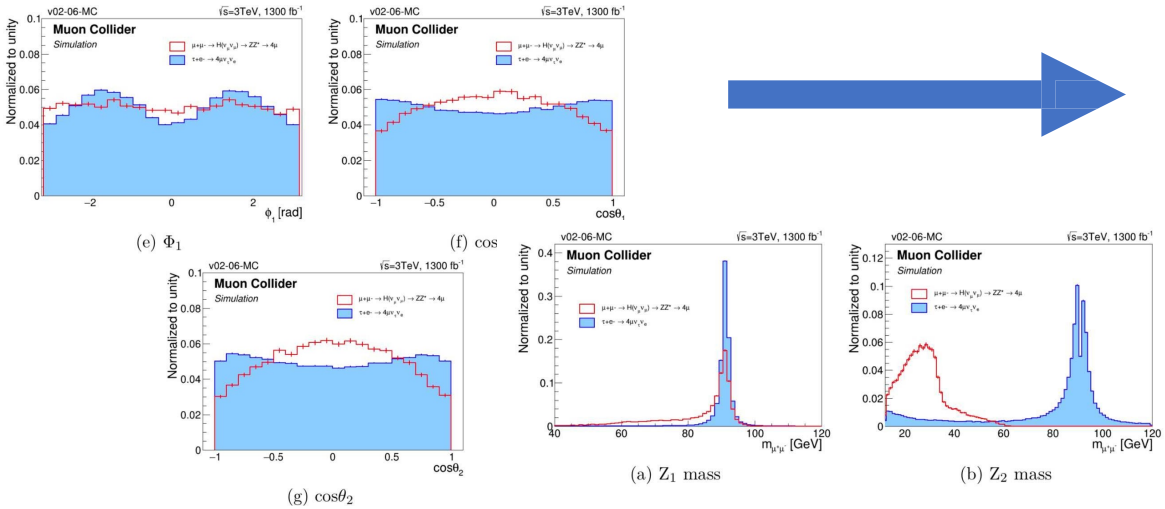
- ML: Features extraction is performed by the user, which choose the discriminating variables to separate signal and background dataset, then the algorithm build the classifier
- DL: the algorithm cluster the events in the hyper-space of N-(unknown) features

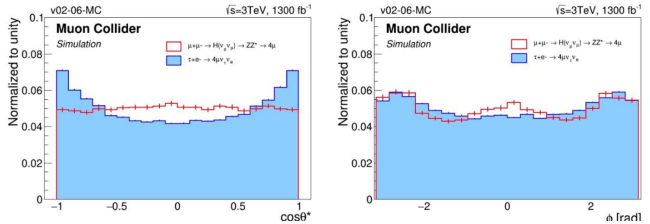
Coming back to our example:

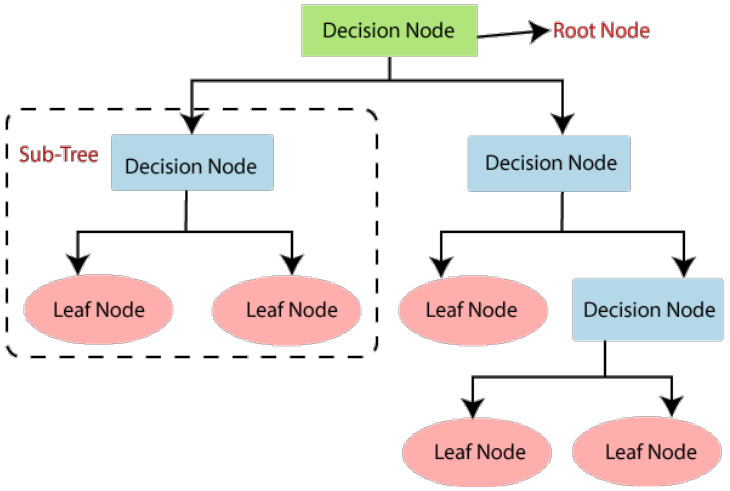
- Signal: 50k events of $\boldsymbol{H \rightarrow ZZ^* \rightarrow 4\mu}$
- Bkg: **200k events of** $\boldsymbol{ZZ \rightarrow 4\mu}$

Input dataset

- Features: Z1 mass, Z2 mass, 5 angular variables
- Algorithm: BDT
- We should create enough examples of signal and background events to feed the BDT → simulate events







How to create a BDT?

Extreme Gradient Boosting

XGBoost

Gradient Boosting

Additive model with loss L:

$$\min_{\alpha_{n=1:N},\beta_{n=1:N}} L\left(y, \sum_{n=1}^{N} \alpha_n f(x, \beta_n)\right)$$

GB approximately solves this objective iteratively and greedily:

$$\min_{\alpha_n,\beta_n} L\left(y, f_{n-1}((x) + \alpha_n f_n(x, \beta_n)\right)$$

XGBoost (A Scalable Tree Boosting System)

Gradient Tree Boosting with Regularization

$$\mathcal{L}(f) = \left[\sum_{i=1}^{n} \ell\left(y^{(i)}, f(x^{(i)})\right)\right] + \gamma J + \frac{1}{2}\lambda \sum_{j=1}^{J} w_j^2$$

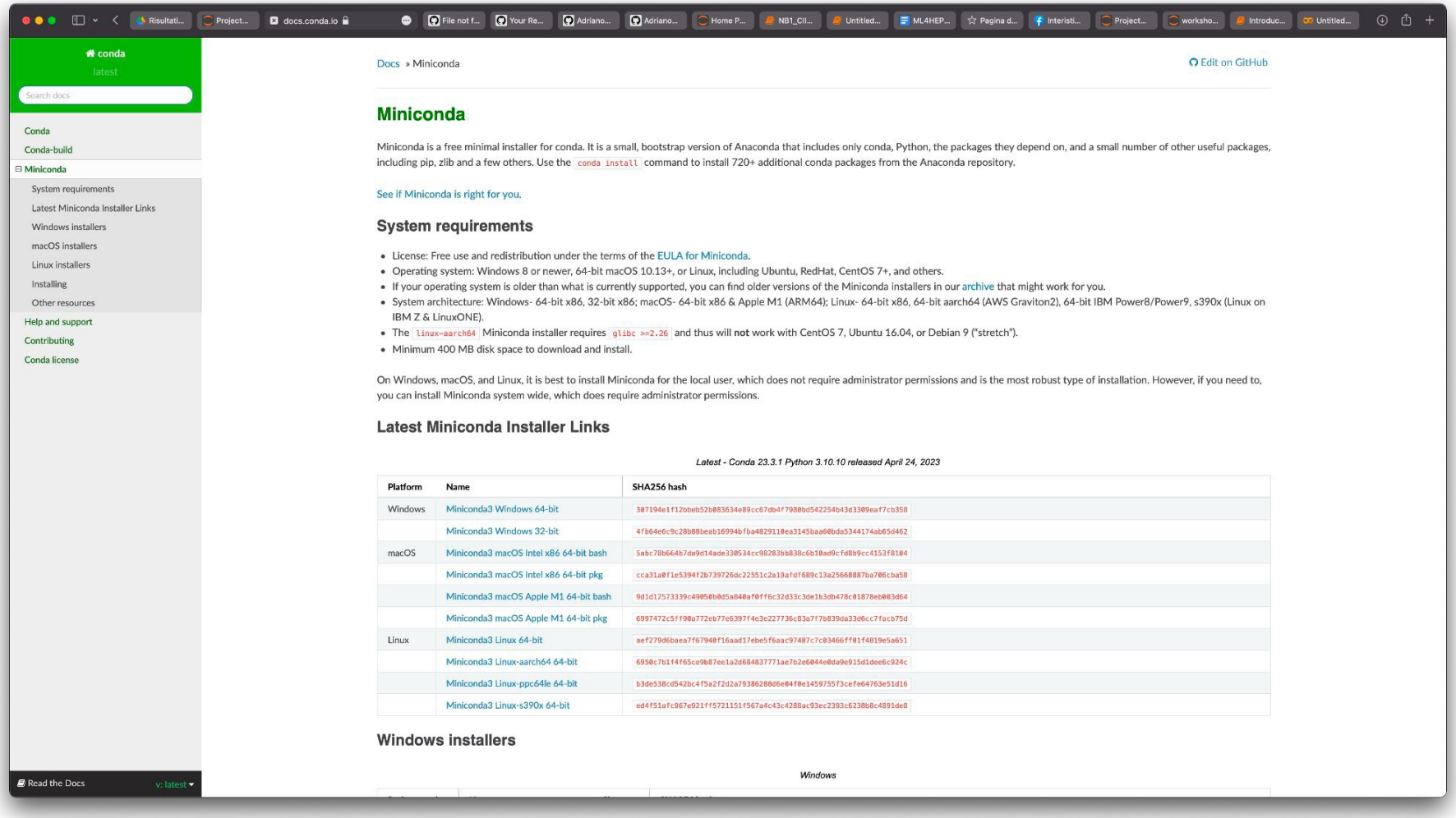
Number of leaves

Prediction associated with the jth leaf

Scalar "hyper-parameters" for regularisation

Setup the environment

1) Install [MiniConda]



Setup the environment

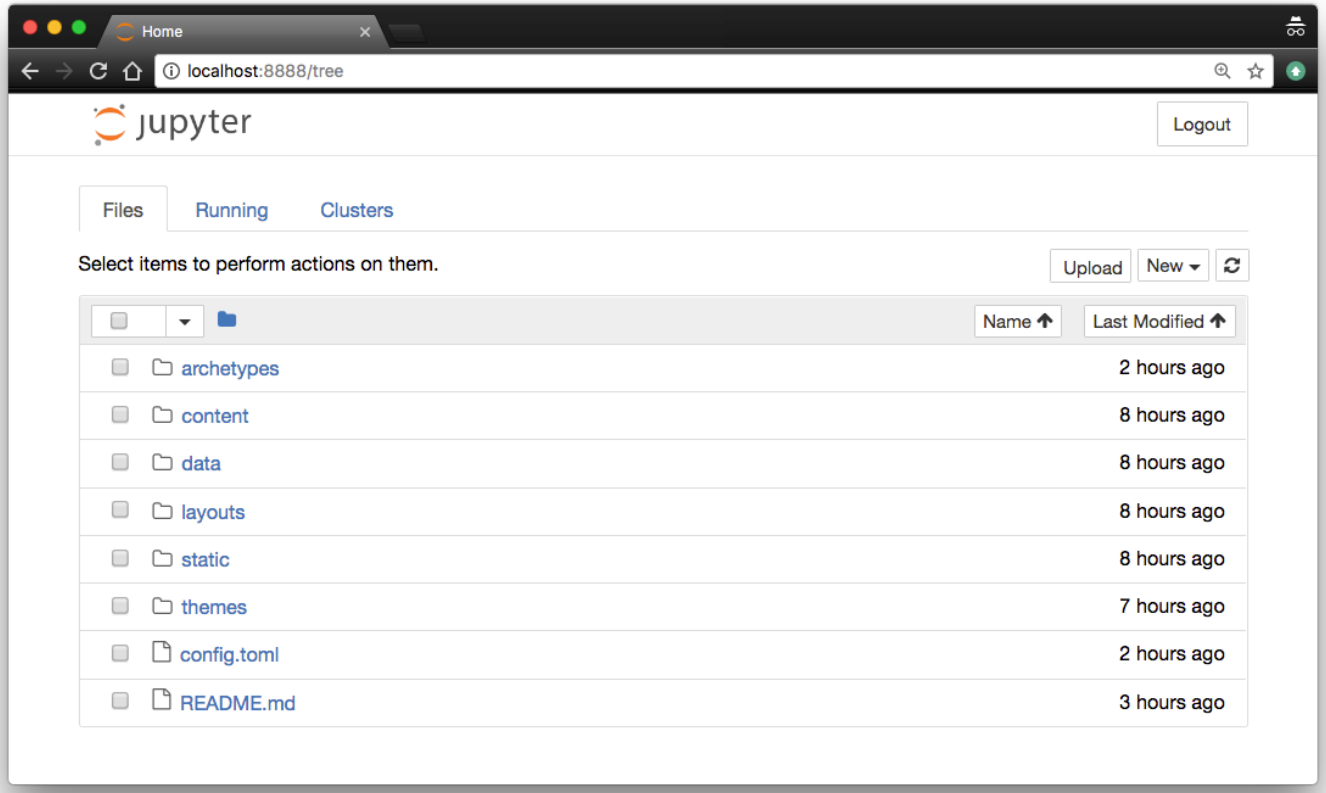
2) Create a conda environment containing ROOT

```
$ conda config --set channel_priority strict
$ conda create -c conda-forge --name <my-environment> root
$ conda activate <my-environment>
```

3) Run jupyter-notebook

```
$ jupyter notebook
```

example entrypoint:
http://**localhost**:8888



Setup the environment

4) Install needed packages

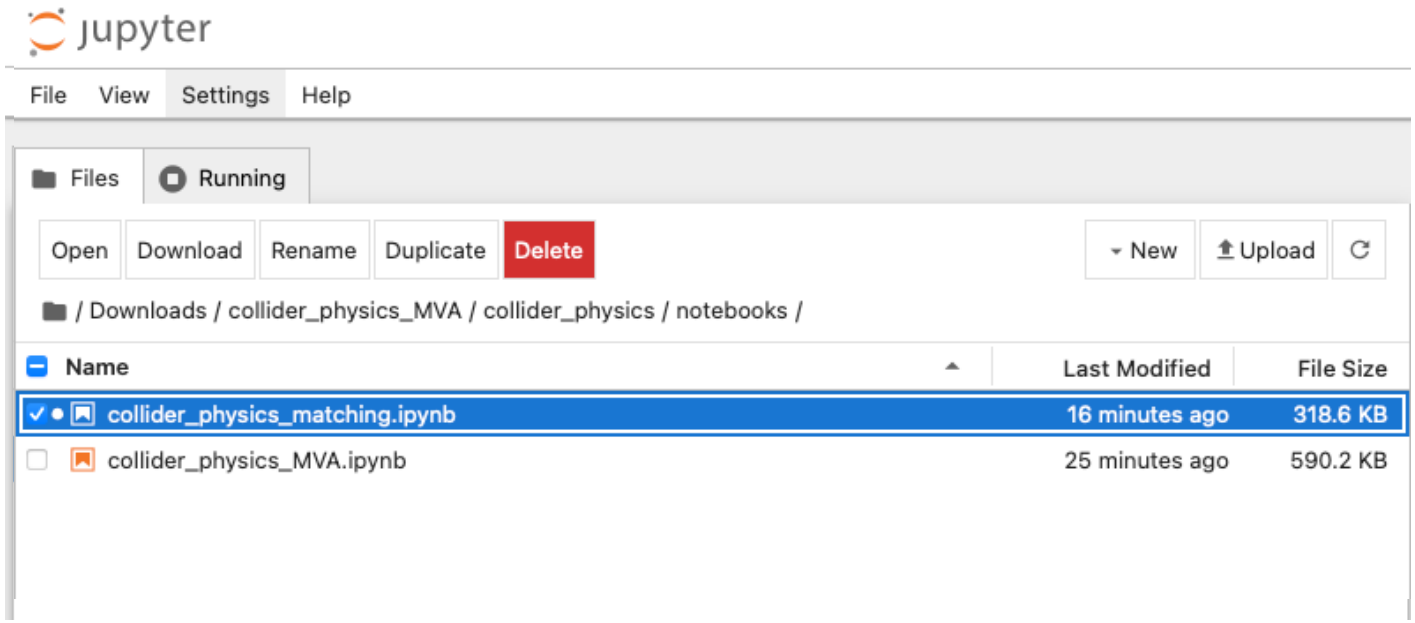
```
$ conda activate <my-environment>
$ pip install pandas
$ pip install matplotlib
$ pip install scikit-learn
$ pip install seaborn
$ pip install --upgrade xgboost

$ conda install wget

$ jupyter notebook
```

Setup the environment

5) Download code

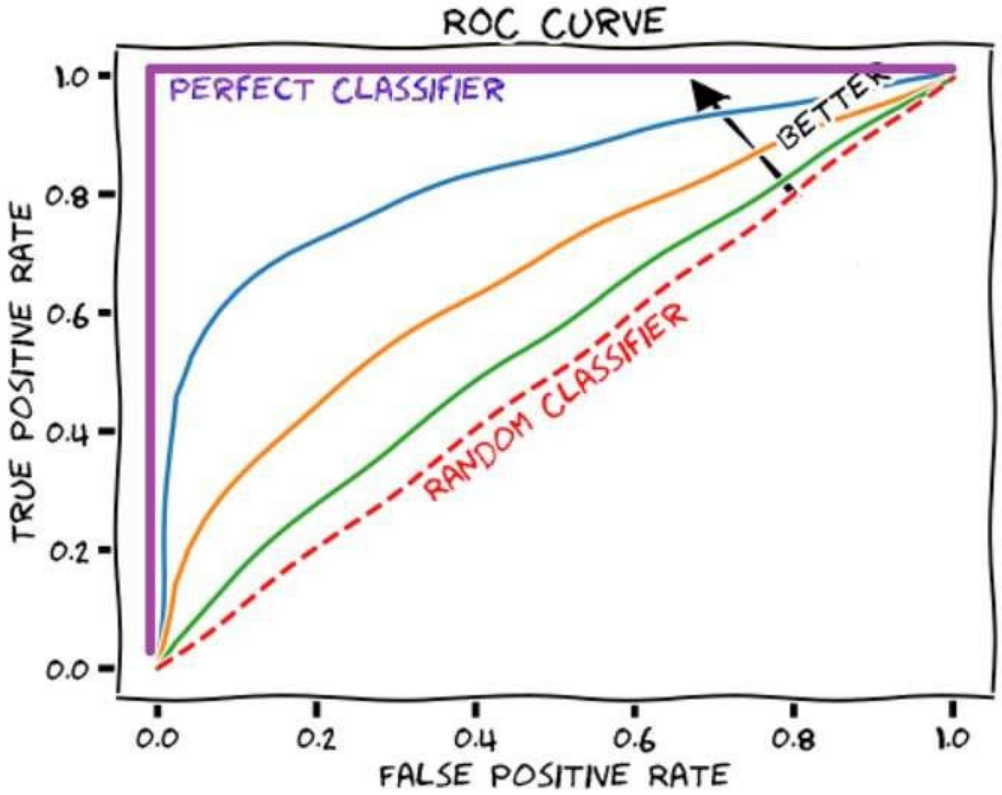
```
$ cd <my_working_directory>
$ git clone
https://github.com/fsimone91/collider_physics.git
```

6) Go to your browser page and open the notebook



Performance of a hypothesis test: the roc curve

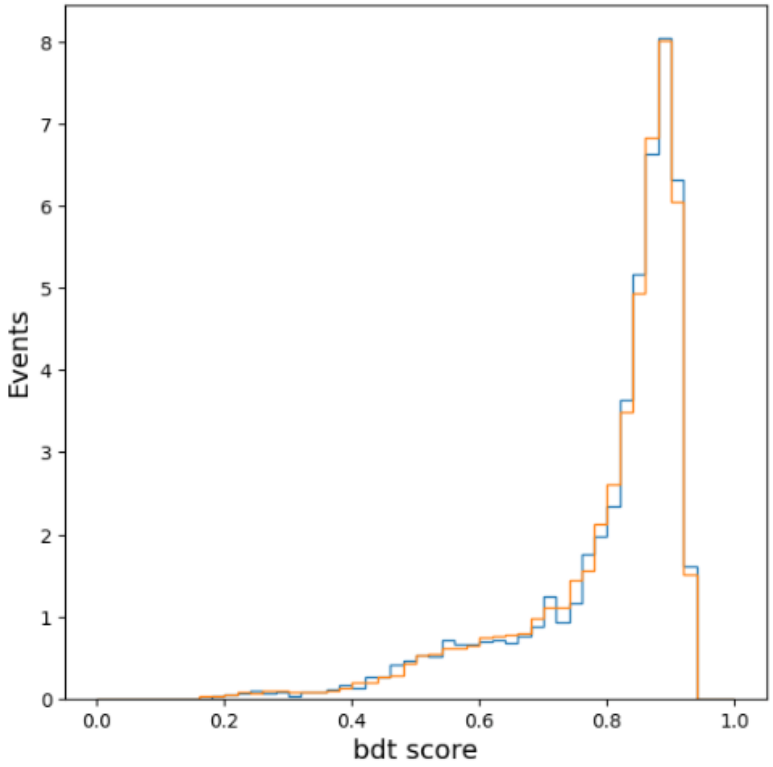
- Being a method to discriminate between 2 hypothesis, the BDT classifier allows to perform an hypothesis test
- The *receiver operating characteristic (ROC) curve is a* typical curve representing
  - the signal efficiency (true positive rate) versus
  - the mis-identification probability (false positive rate, that is the probabability that akg is classified as signal) obtained by varying the selection requirement



- The turn on of the roc curve allows to choose the selection to be applied on the classifier to discrminate signal and background
- A good selection should have a low misidentification probability corresponding to a large selection efficiency →Perfect classifier should have the pdf such that
  - Efficiecy = 1
  - Mis-id rate=0

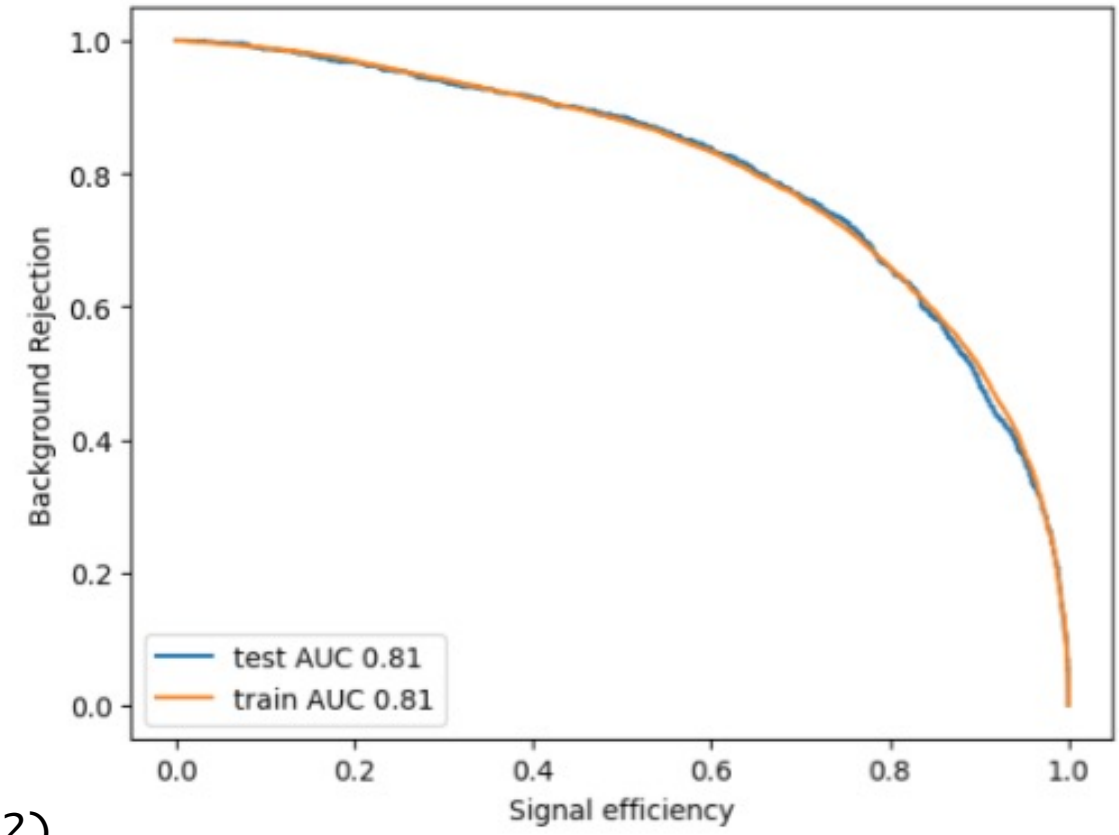
Overtraining Checks

- Kolmogorov test quantifies the agreement between test and training histograms



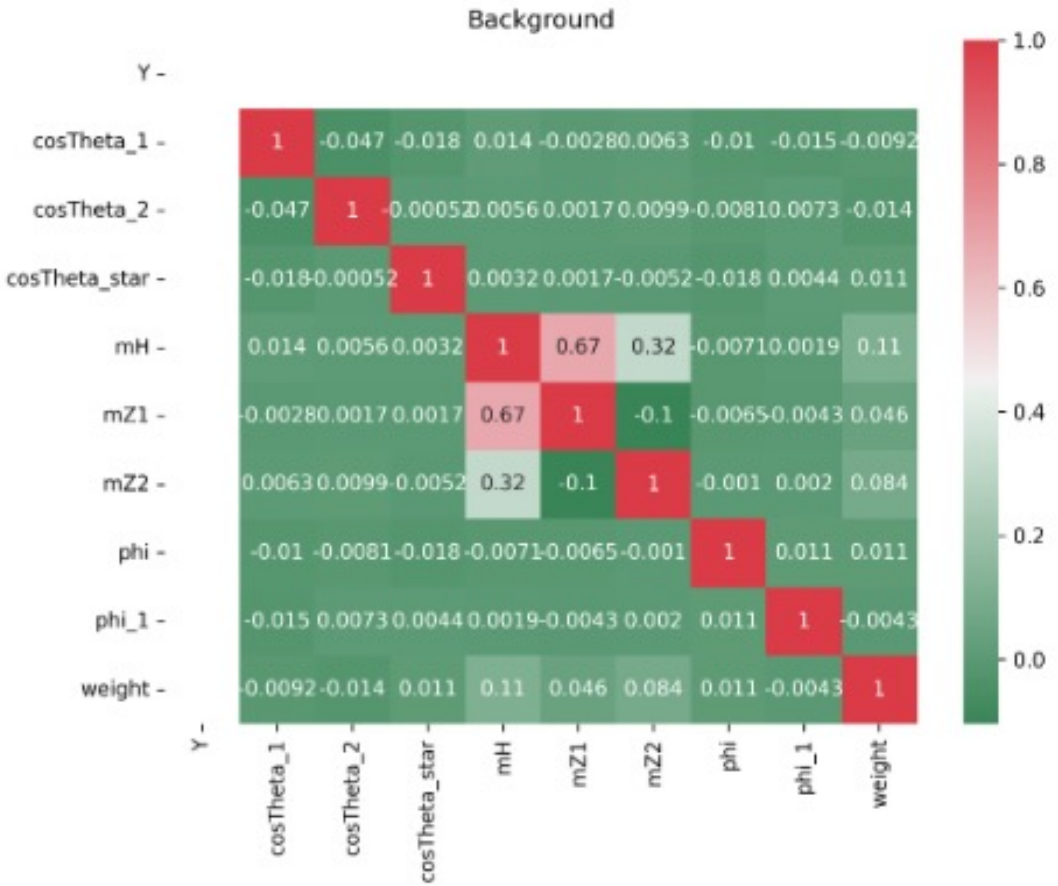

```
KstestResult(statistic=0.0152,
pvalue=0.51, statistic_location=0.82)
```

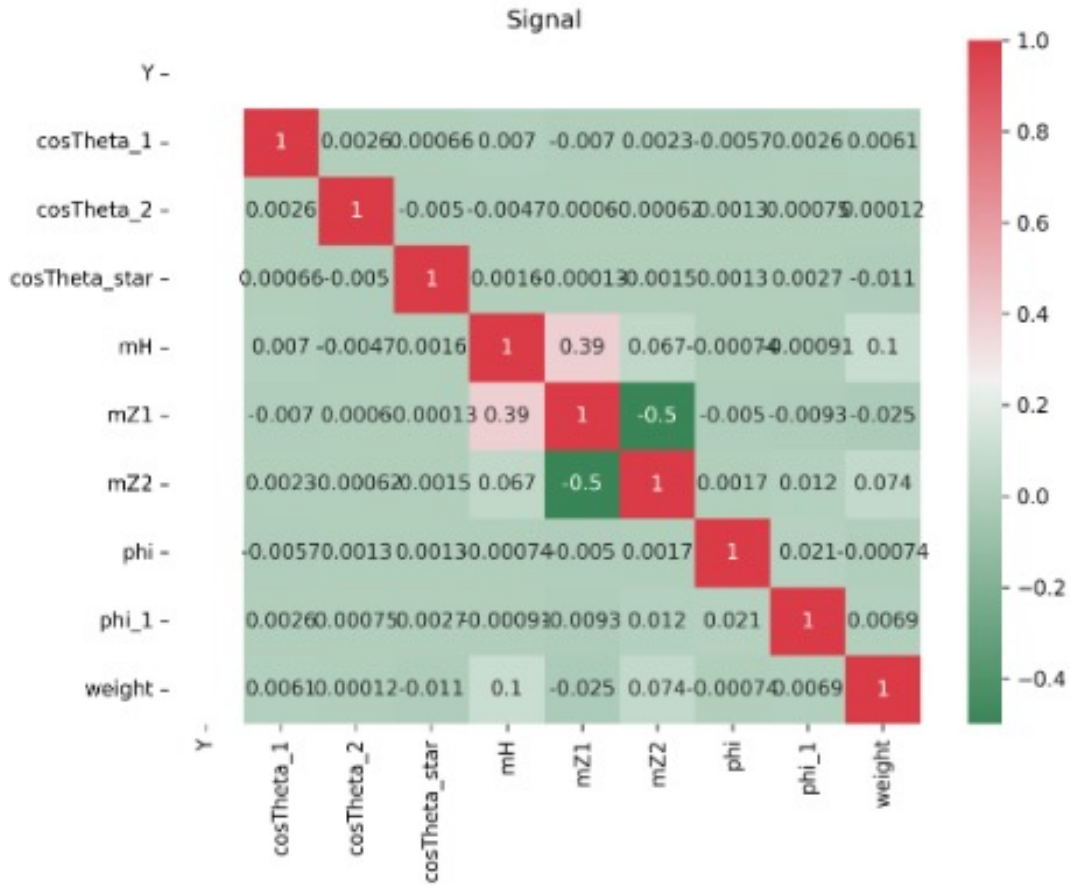
- The ROC curves for test and training is one of the possible ways to qualify the generalisation power of our discriminator



Correlation matrix

- Differences in correlation between features in signal and background are exploited by the algorithm that build the classifier
- If there are too many correlations among the input features, it is a symptom of the fact that we are using redundant information and we could obtain the same results using less variables





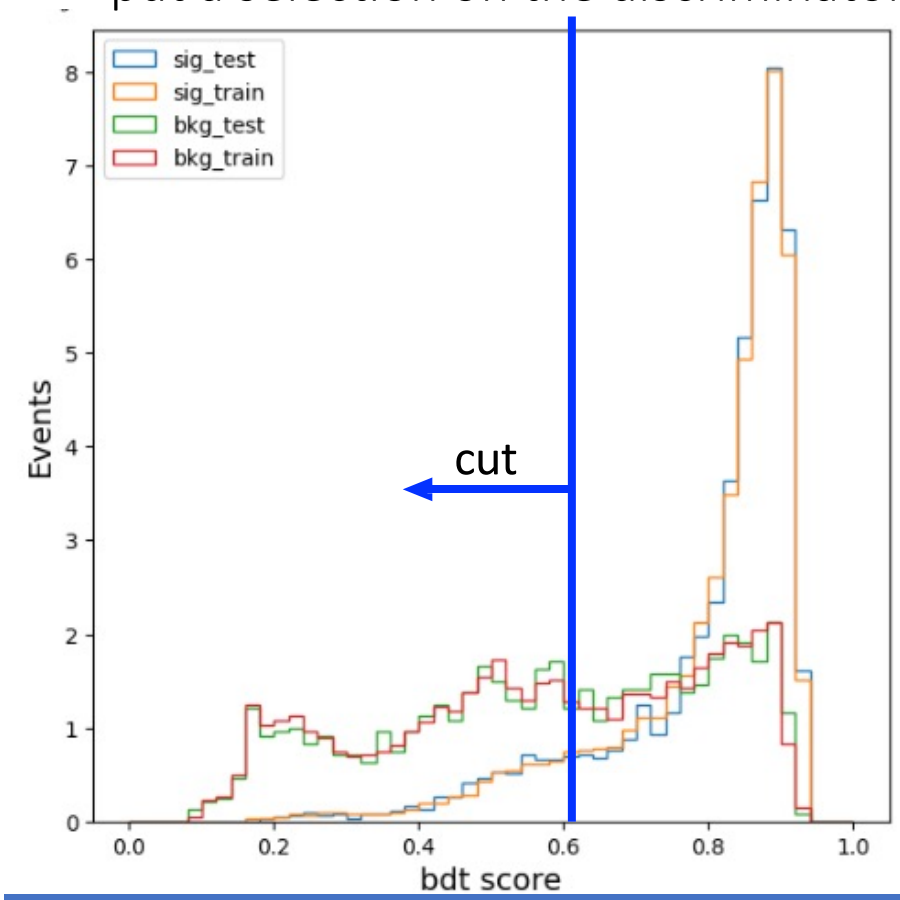
How to integrate MVA in an analysis?

Next step: apply on unknown, never seen, data

- The discriminator you built is just a function to be evaluated on data
- The functional form (basically the weights associated to the output of each neuron/layer/decision tree) dependend on the input variables and parameters set that you have used in the training phase
- Everything is stored in files (e.g. xml) that can be read by the MVA library (xgboost, pytorch, tensorflow, root TMVA etc) in order to evaluate the function on data → **inference**

How to integrate MVA in an analysis?

- Let's suppose we computed the classifier on new data (coming from real collisions), where signal (H->ZZ→4μ) and SM background (ZZ →4μ) are mixed
- Our goal: discriminate signal events from background events
- We can exploit the MVA classifier to do the discrimination → this means we will put a selection on the discriminator value computed in data



Data Events with classifier values below the selection threshold are background like--> we will reject them

After BDT selection

aka: selection on never seen data

In this case, we apply all the selections to new generated MC samples (**not the ones used for training!!!**!), to check the effect

significance

$$Z = \sqrt{2((S+B)\ln(1+\frac{S}{B}) - S)}$$

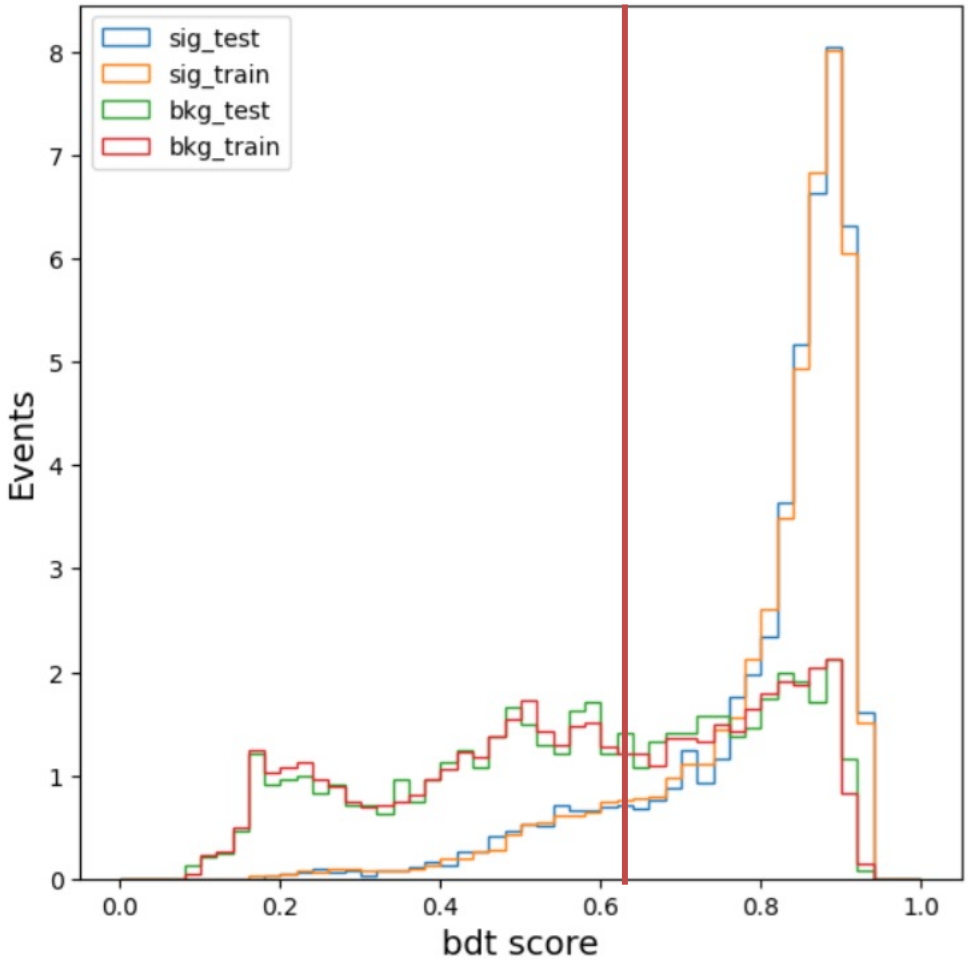
$\sqrt{s} = 1.5$ TeV and $L = 500$ fb^{-1}					
Cathegory	Signal		Background		Z
	Events	Efficiency (%)	Events	Efficiency (%)	
Generated	4.57	–	4.59	–	–
Pre-selected	2.98±0.02	65.12±0.21	3.93±0.01	85.76±0.08	1.35
After BDT cut	2.96±0.02	64.71±0.21	0.57±0.01	12.50±0.07	2.63
Higgs mass range	2.88±0.02	63.11±0.22	0.13±0.01	2.73±0.04	3.66

$\sqrt{s} = 3$ TeV and $L = 1300$ fb^{-1}					
Cathegory	Signal		Background		Z
	Events	Efficiency (%)	Events	Efficiency (%)	
Generated	19.16	–	23.21	–	–
Pre-selected	10.77±0.06	56.18±0.22	18.57±0.05	80.04±0.09	2.30
After BDT cut	10.70±0.06	55.85±0.22	2.37±0.02	10.22±0.07	4.82
Higgs mass range	10.44±0.06	54.46±0.22	0.47±0.01	2.01±0.03	6.91

Exercise

(https://github.com/fsimone91/collider_physics/blob/main/notebooks/collider_physics_MVA.ipynb)

Goal: find the optimal BDT score value to cut on



Procedure:

- Scan over the BDT score on the test set
- For each BDT score value, count signal events and background events passing the selection
- Compute the signal significance and store it:

$$Z = \sqrt{2((S+B)\ln(1+\frac{S}{B}) - S)}$$

- Find the BDT score value corresponding to the maximum value of the signal significance