Signal/background discrimination for the VBF Higgs four lepton decay channel with the CMS experiment using Machine Learning classification techniques

B. D'Anzi – N. De Filippis – D. Diacono – G. Miniello – W. Elmetenawee – A. Sznajder
Università e INFN BARI

Link

Tirocinio INFN- Sez.Bari – Laurea Magistrale in Nuclear, SubNuclear and Astroparticle Physics

Author(s)

Name	Institution	Mail Address	Social Contacts
Brunella D'Anzi	INFN Sezione di Bari	brunella.d'anzi@cern.ch	Skype: live:ary.d.anzi_1; Linkedin: brunella-d-anzi
Nicola De Filippis	INFN Sezione di Bari	nicola.defilippis@ba.infn.it	
Domenico Diacono	INFN Sezione di Bari	domenico.diacono@ba.infn.it	
Walaa Elmetenawee	INFN Sezione di Bari	walaa.elmetenawee@cern.ch	
Giorgia Miniello	INFN Sezione di Bari	giorgia.miniello@ba.infn.it	
Andre Sznajder	Rio de Janeiro State University	sznajder.andre@gmail.com	

Supervisors

Tested on:
- Colab

From ownCloud RecasBari

How to execute it

Use Googe Colab

What is Google Colab?

Google's Colaboratory is a free online cloud-based Jupyter notebook environment on Google-hosted machines, with some added features, like the possibility to attach a GPU or a TPU if needed with 12 hours of continuous execution time. After that, the whole virtual machine is cleared and one has to start again. The user can run multiple CPU, GPU, and TPU instances simultaneously, but the resources are shared between these instances.

Open the Use Case Colab Notebook

The notebook for this tutorial can be found **here**. The .ipynb file is available in the **Attachment section** and in this **GitHub repository**.

Indeed, the notebook can be opened by inserting the GitHub URL **bdanzi/Higgs_exercise** and clicking on the **VBF_exercise.ipynb** icon :

Attachment section

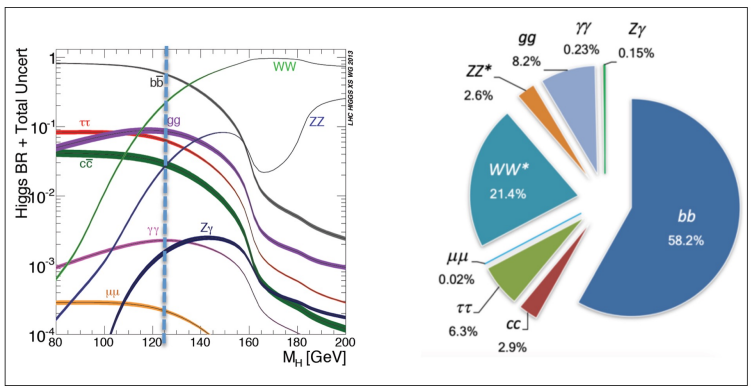
From GitHub

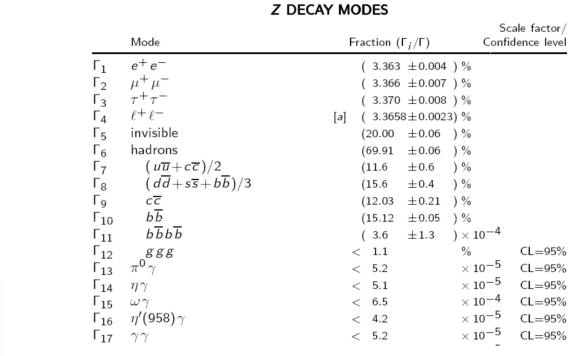
From URL

OR one can just click on the following link: **https://colab.research.google.com/drive/1hVA0E5kosM2gdFkJINb6WeVp5hjG1ML1?usp=sharing.**

Be sure to work on a **copy** of the notebook in Google Drive in both cases clicking on the **Copy to Drive** icon as shown below:

Cosa fa?





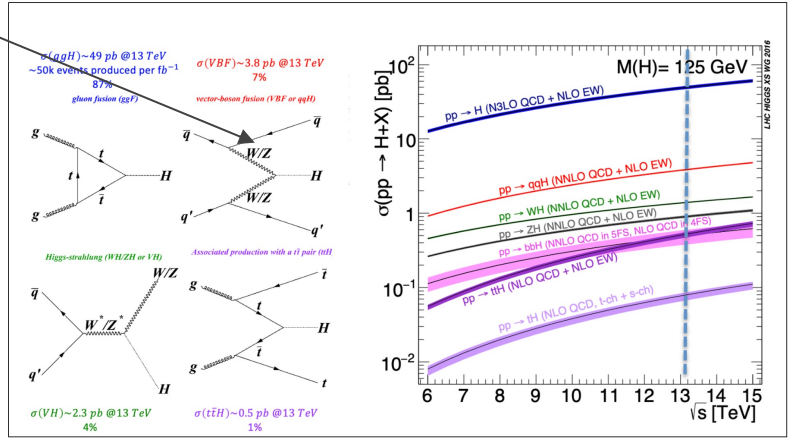
Z DECAY MODES

	Mode	Fraction (Γ_i/Γ)	Scale factor/ Confidence level
Γ_1	e^+e^-	(3.363 ±0.004) %	
Γ_2	$\mu^+\mu^-$	(3.366 ±0.007) %	
Γ_3	$\tau^+\tau^-$	(3.370 ±0.008) %	
Γ_4	$\ell^+\ell^-$	[a] (3.3658±0.0023) %	
Γ_5	invisible	(20.00 ±0.06) %	
Γ_6	hadrons	(69.91 ±0.06) %	
Γ_7	$(u\overline{u}+c\overline{c})/2$	(11.6 ±0.6) %	
Γ_8	$(d\overline{d}+s\overline{s}+b\overline{b})/3$	(15.6 ±0.4) %	
Γ_9	$c\overline{c}$	(12.03 ±0.21) %	
Γ_{10}	$b\overline{b}$	(15.12 ±0.05) %	
Γ_{11}	$b\overline{b}b\overline{b}$	(3.6 ±1.3) $\times 10^{-4}$	
Γ_{12}	ggg	< 1.1 %	CL=95%
Γ_{13}	$\pi^0\gamma$	< 5.2 $\times 10^{-5}$	CL=95%
Γ_{14}	$\eta\gamma$	< 5.1 $\times 10^{-5}$	CL=95%
Γ_{15}	$\omega\gamma$	< 6.5 $\times 10^{-4}$	CL=95%
Γ_{16}	$\eta'(958)\gamma$	< 4.2 $\times 10^{-5}$	CL=95%
Γ_{17}	$\gamma\gamma$	< 5.2 $\times 10^{-5}$	CL=95%

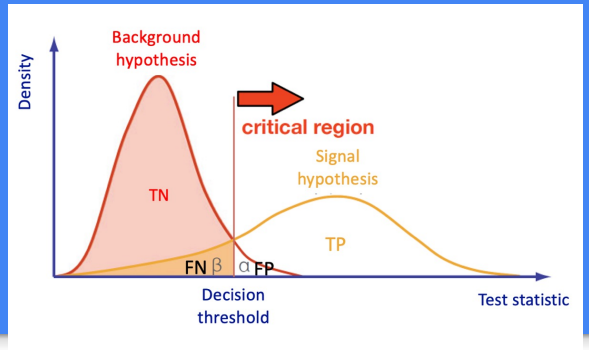
- Selezione di eventi di produzione di singolo bosone di Higgs (ipotesi di massa 125 GeV) tramite il meccanismo di vector boson fusion (VBF) con stato finale in 4 muoni (canale 4e - esercizio opzionale) @ CMS (datasets al generator level) ;

- Alcuni dei background irriducibili : gg->H->ZZ->4mu, QCD ZZ->4mu, ttbarH->ZZ->4mu (esercizio opzionale, basta togliere il commento a qualche riga di codice nello use-case) ;

- Il segnale si presenta come «processo raro», difficile da discriminare dal fondo con tecniche di analisi multivariata standard (imponendo dei tagli sulle osservabili fisiche) -> Utilizziamo algoritmi di Machine Learning per la nostra analisi multivariata!



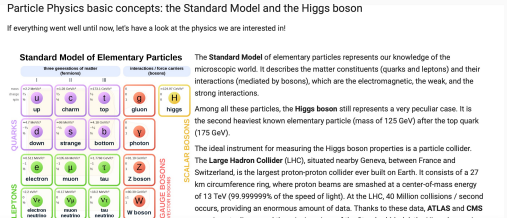
Lo use-case



- Un notebook. In vista dell' hackathon si potrebbe pensare di implementare solo la Artificial Neural Network (ANN) o Random Forest (RF).
- È suddiviso in sezioni :

1. **Introduzione (Standard Model, Hypothesis test per discriminare tra segnale e rumore) – gli esperti possono saltare questa parte**
    - accesso a ~114k eventi mergiati tra segnale (14k) e rumore (100k) da ROOT files.

Sommario


Particle Physics basic concepts: the Standard Model and the Higgs boson

If everything went well until now, let's have a look at the physics we are interested in!

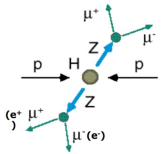
Standard Model of Elementary Particles

The **Standard Model** of elementary particles represents our knowledge of the microscopic world. It describes the matter constituents (quarks and leptons) and their interactions (mediated by bosons), which are the electromagnetic, the weak, and the strong interactions.

Among all these particles, the **Higgs boson** still represents a very peculiar case. It is the second heaviest known elementary particle (mass of 125 GeV) after the top quark (175 GeV).

The ideal instrument for measuring the Higgs boson properties is a particle collider. The **Large Hadron Collider** (LHC), situated nearby Geneva, between France and Switzerland, is the largest proton-proton collider ever built on Earth. It consists of a 27 km circumference ring, where proton beams are smashed at a center-of-mass energy of 13 TeV (99.999999% of the speed of light). At the LHC, 40 Million collisions / second occurs, providing an enormous amount of data. Thanks to these data, **ATLAS** and **CMS**

Pictures from the notebook



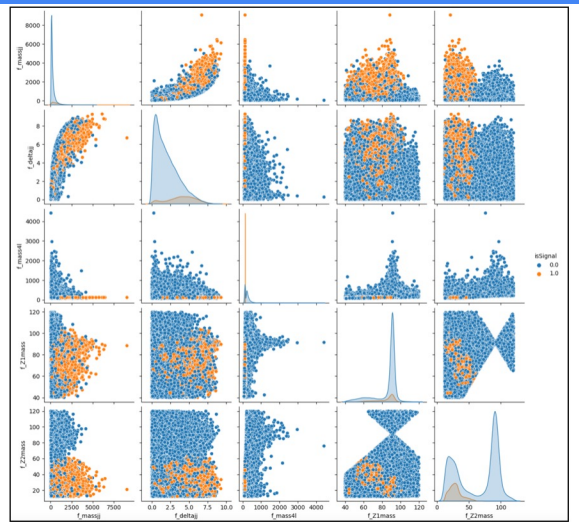
Our physics problem consists in detecting the so-called **"golden decay channel"** $H \to ZZ^* \to l^+l^-l'^+l'^-$ which is one of the possible Higgs boson's decays: its name is due to the fact that it has the clearest and cleanest signature of all the possible Higgs boson's decay modes. The decay chain is sketched here: the Higgs boson decays into Z boson pairs, which in turn decay into a lepton pair (in the picture, muon-antimuon or electron-positron pairs). In this exercise, we will use only datasets concerning the 4μ **decay channel** and the datasets about the **4e channel** are given to you to be analyzed as an optional exercise. At the LHC experiments, the decay channel **2e2mu** is also widely analyzed.

Lo use-case

Load data using PANDAS data frames

Preparing input features for the ML algorithms

Dividing the data into testing and training dataset



2. Caricamento dati

- Evidenziare il problema della scelta delle features per la fase di training dei nostri algoritmi di Machine Learning : si sono prodotti alcuni plot che mostrano la correlazione tra le variabili. La scelta ricadeva su high-level o low level features. Si scelgono quelle con cui gli algoritmi mostrano prestazioni migliori (check fatto preliminarmente da noi).

Let's print out the first rows of this dataset!

```
df['sig'].head()
```

	f_run	f_event	f_weight	f_massjj	f_deltajj	f_mass4l	f_Z1mass	f_Z2mass
0	1	385228	0.000176	667.271423	3.739947	124.966576	90.768616	20.508274
1	1	385233	0.000127	129.085892	0.046317	120.231926	80.782318	34.261726
2	1	385254	0.000037	285.165222	3.166899	125.254646	91.392693	25.695290
3	1	385260	0.000043	52.006794	0.150803	125.067009	91.183708	19.631315
4	1	385263	0.000092	1044.083496	4.315164	124.305748	72.480515	43.826504

```
df['sig'].head()
```

	f_lept1_pt	f_lept1_eta	f_lept1_phi	f_lept2_pt	f_lept2_eta	f_lept2_phi	f_lept3_pt	f_lept3_eta	f_lept3_phi	f_lept4_
	82.890457	0.822203	1.343706	65.486946	0.382922	2.568485	39.838531	0.546917	2.497204	28.562
	41.195362	-0.534245	2.802684	24.911942	-2.065928	0.371150	21.959597	-1.219900	-2.938914	16.676
	80.788002	0.943778	0.729632	35.549721	0.935241	1.288549	23.206284	0.236346	-2.670540	14.581
	129.883423	0.235406	-1.729384	37.950790	1.226075	-2.540356	17.678413	0.096546	-1.533120	8.197
	86.220734	-0.226653	0.117277	80.451378	-0.536749	0.385678	27.497240	0.827591	-0.072236	21.243

Lo use-case

3. **Descrizione e implementazione di una Artificial Neural Network (ANN) e una Random Forest**
   - Sono presenti parti teoriche di base che gli esperti possono saltare.
   - Si utilizzano Scikit-learn e Keras API

Description of the Artificial Neural Network (ANN) model and KERAS API

Introduction to the Neural Network algorithm

Usage of Keras API: basic concepts

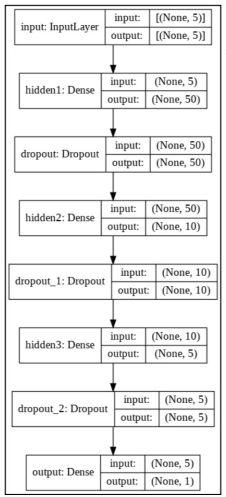
Artificial Neural Network implementation

Description of the Random Forest (RF) and Scikit-learn library

Introduction to the Random Forest algorithm

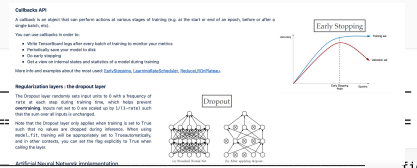
Optional exercise : Draw a decision tree

Random Forest implementation

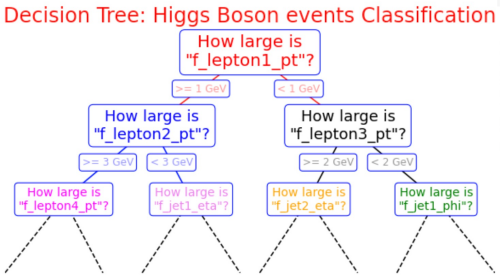


ANN

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input (InputLayer)           (None, 5)                 0
_________________________________________________________________
hidden1 (Dense)              (None, 50)                300
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
hidden2 (Dense)              (None, 10)                510
_________________________________________________________________
dropout_1 (Dropout)          (None, 10)                0
_________________________________________________________________
hidden3 (Dense)              (None, 5)                 55
_________________________________________________________________
dropout_2 (Dropout)          (None, 5)                 0
_________________________________________________________________
output (Dense)               (None, 1)                 6
=================================================================
Total params: 871
Trainable params: 871
Non-trainable params: 0
_________________________________________________________________
```




```
fig.savefig('05.08-decision-tree.png')
```




```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import plot_roc_curve
    from sklearn.model_selection import GridSearchCV
```

Let's implement the grid search algorithm for our Random Forest discriminator!

Grid Search algorithm basically tries all possible combinations of parameter values and returns the combination with the **highest accuracy**. The Grid Search algorithm can be very slow, owing to the potentially huge number of combinations to test. Furthermore, performing **cross-validation** considerably increases the execution time of the process!
For these reasons, the algorithm is commented on the following code cells and images of the outputs are left to you!

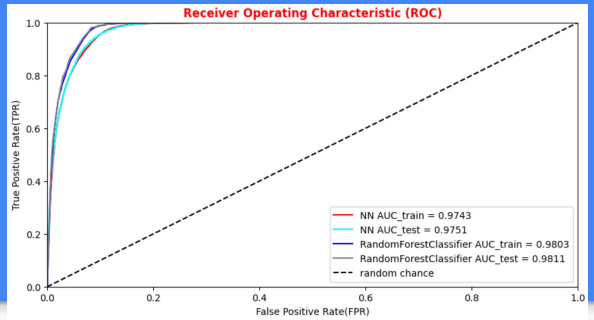
To read more about cross-validation on Scikit-learn:

Cross_validation

To read more about GridSearchCV algorithm on Scikit-learn:

GridSearchCV algorithm

Lo use-case



Performance evaluation

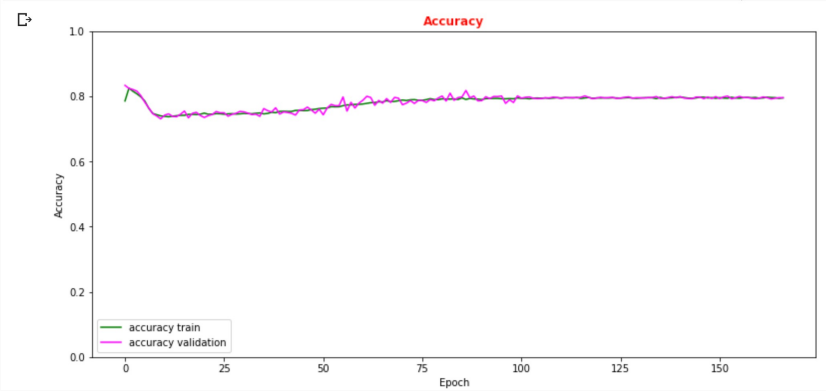
ROC curve and rates definitions

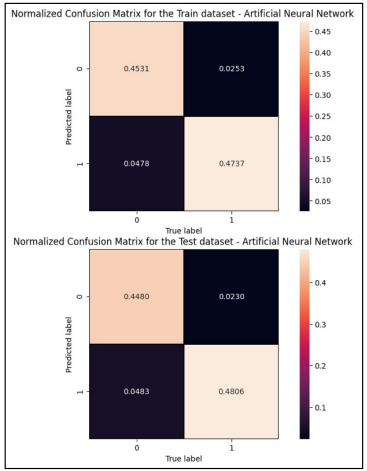
Overfitting and test evaluation of an MVA model

Artificial Neural Network performance

Exercise 1 - Random Forest performance

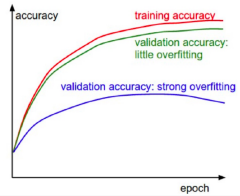
3. **Valutazione quantitativa di quanto hanno imparato dal training i nostri algoritmi**
   - Qualche plot e metrica utilizzata
   - Esercizio: valutazione delle performance per la RF (codice simile all'ANN).

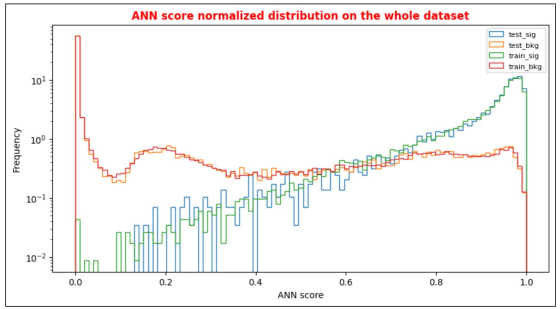




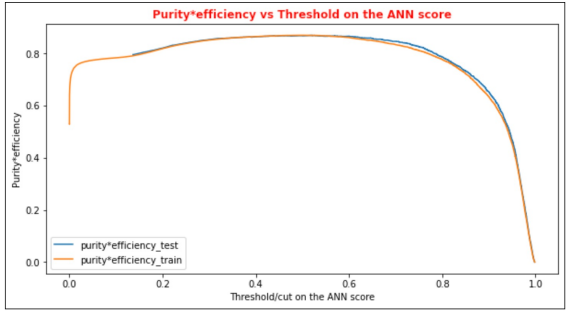
Overfitting and test evaluation of an MVA model

The loss function and the accuracy metrics give us a measure of the **overtraining (overfitting)** of the ML algorithm. Over-fitting happens when an ML algorithm learns to recognize a pattern that is primarily based on the training (validation) sample and that is nonexistent when looking at the testing (training) set (see the plot on the right side to understand what we would expect when overfitting happens).





Lo use-case



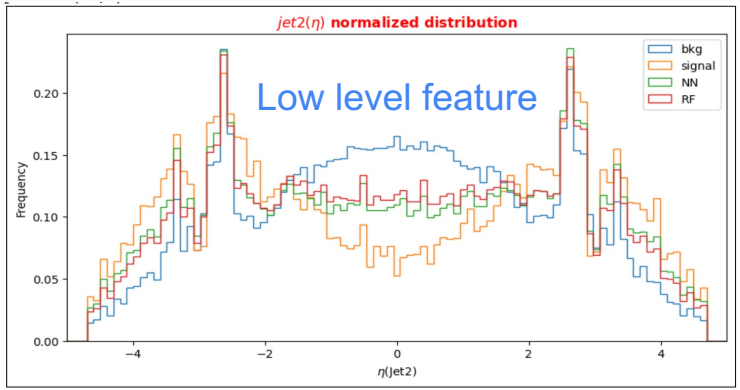
Plot physics observables

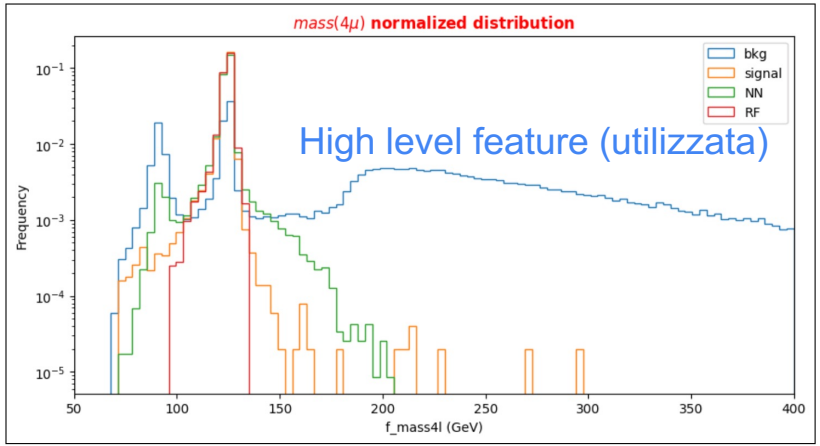
Artificial Neural Network rates fixing an ANN score threshold from data frame

Exercise 2 - Random Forest rates fixing a RF score threshold from data frame

Plot some physical quantities after that the event selection is applied

3. **Selezione di eventi di segnale applicando un taglio sullo score dell'ANN / Random Forest guardando alla metrica purity*efficiency**
   - Qualche plot relativo a variabili fisiche utilizzate e non nel training degli algoritmi
   - Esercizio: procedura di selezione degli eventi per la RF usando data frames (codice simile all'ANN).





Lo use-case

3. **Machine Learning challenge**
   - Sono proposti esercizi di inclusione di un terzo background ttbarH
   - Esercizio: migliorare uno dei due modelli a scelta (ANN o RF) per la discriminazione segnale vs background (i due mergiati) nel canale 4 muoni (o 4 elettroni - da inserire) di un nuovo dataset di cui non si conosce la flag isSignal e caricare le y_pred sull'ownCloud Recas (noi usiamo un notebook a parte che calcola AUC della ROC Curve). Chi ottiene una AUC maggiore presenta il proprio modello!

```
#This is selecting the full dataset, it will take a while (2-3 minutes).
#Comment afterwards, no need to re-run this box!

!pip install uproot

files = { #time for uploading = almost 3 minutes
    "VBF_HToZZTo4mu.root" : "PCH8ZgVPVwgtqXF", #signal events| 4-muons channel
    "GluGluHToZZTo4mu.root": "64ngyPZosPKsp3Q", #1st background events| 4-muons ch
    "ZZTo4mu.root"       : "JUzjDb5tjy6tZpC", # 2nd background| 4-muons
    #"ttH_HToZZ_4mu.root"   :"RzS4CXKExWijjfQ", 3nd background | 4-muons
    #"VBF_HToZZTo4e.root": "dBdaKwA13bygU8c", # alternative signal| 4e-channel
    #"GluGluHToZZTo4e.root": "2DxFLqLtjkbWFdM", #alternative background| 4e-ch
    #"ZZTo4e.root"     : "NhAutemXTrVunU0", #alternative background| 4e-ch
     #"ttH_HToZZ_4e.root" :   "DRDeVWSSZzJNduw" #alternative background| 4e-ch
    }
!rm -f *.root
import os
for file in files.items():
  if not os.path.exists(file[0]):
    b = os.system ( "wget -O %s --no-check-certificate 'https://recascloud.ba.infn.it/index.php
    if b: raise IOError ( "Error in downloading the file %s : (%s)" % file )
```

Optional Exercise 1 - Change the decay channel

Optional Exercise 2 - Merge the backgrounds

Machine Learning challenge

```
# Converting the dataframe into a csv file
# Modify the 'answer.csv' string in the line code below and insert your name and the ML model trained (rf or nn)!
# Example: df_answer.to_csv('mario_rossi_rf_4mu.csv')
df_answer.to_csv('answer_2017_trial.csv')
print('Your y_pred has been created! Download it from your Drive directory!\n')
!ls -l
```

```
Your y_pred has been created! Download it from your Drive directory!

total 12884
-rw-r--r-- 1 root root   37013 Apr 26 09:11 05.08-decision-tree.png
-rw-r--r-- 1 root root   46864 Apr 26 09:11 ANN_model.h5
-rw-r--r-- 1 root root 2981984 Apr 26 09:13 answer_2017_trial.csv
-rw-r--r-- 1 root root 9160896 Apr 26 09:13 input_hl.csv
-rw-r--r-- 1 root root   44288 Apr 26 09:09 model.png
-rw-r--r-- 1 root root  902391 Apr 26 09:12 rf_model.pkl
drwxr-xr-x 1 root root    4096 Apr 21 13:39 sample_data
```

Upload your results here:

https://recascloud.ba.infn.it/index.php/s/CnoZuNrlr3x7uPl

Su INFN Confluence

General Information

ML/DL Technologies	Artificial Neural Networks (ANNs), Random Forests (RFs)
Science Fields	High Energy Physics
Difficulty	Low
Language	English
Type	fully annotated and runnable

Software and Tools

Programming Language	Python
ML Toolset	Tensorflow, Keras, Scikit-learn
Additional libraries	uproot, NumPy, pandas,h5py,seaborn,matplotlib
Suggested Environments	Google's Colaboratory

Needed datasets

Data Creator	CMS Experiment
Data Type	Simulation
Data Size	2 GB
Data Source	Cloud@ReCaS-Bari

Attachments

Here it is the complete notebook:

VBF_H_4l_ML_exercise.ipynb