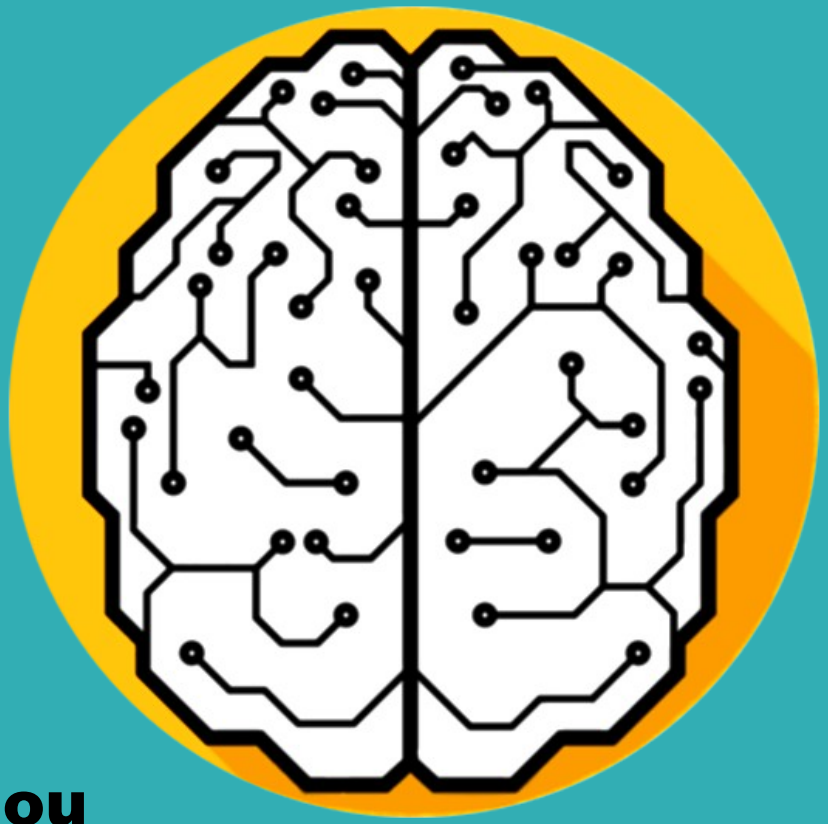
Parallelization of ROOT Machine Learning Methods



Pourya Vakilipourtakalou

Supervisors : Prof. Lorenzo Moneta

Prof. Sergei Gleyzer

Overview

- Machine Learning
- ROOT
- TMVA
- Cross Validation
- Parallelization
- Outlook

Machine Learning

Teaching the computers to do something exactly like the way people learn.

How do people learn?

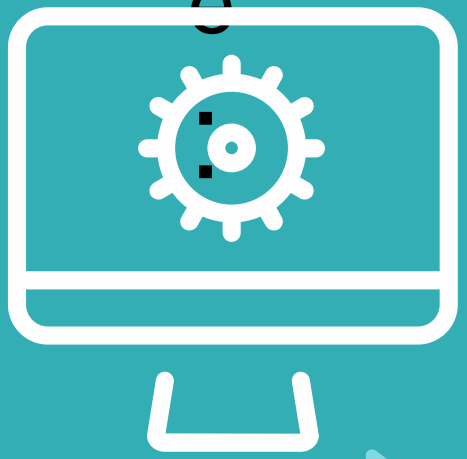
H o r s e :



And this is Machine Learning!

Cake:





Data

Algorithm

Output

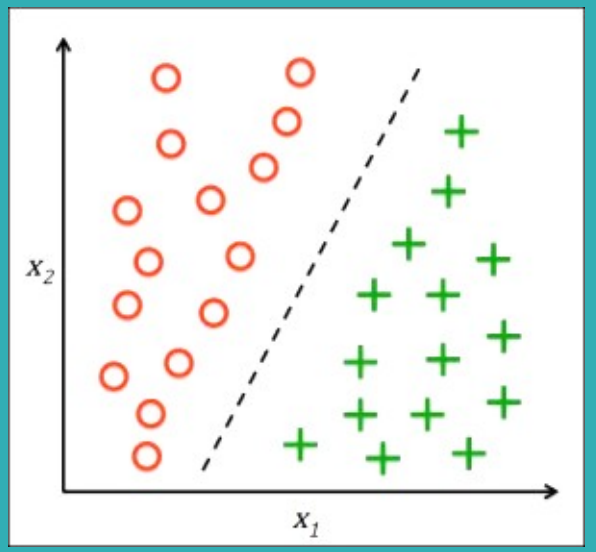
Data

Output

Trained Algorithm

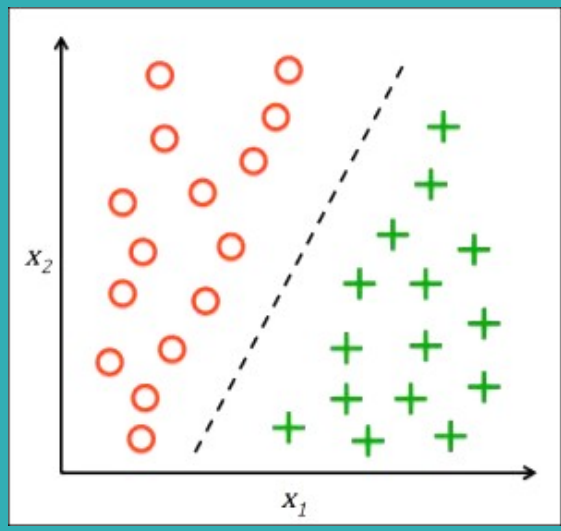
We tra

Machine Learning



- An example of classification
- X1 → age of the patient
- X2 → size of the tumor
- Y → output : Malignant or Benign → 0 or 1
- Proposing a function like H(X1,X2) like
  aX1 + bX2 → it can be anything
- Try to find optimal a and b

Machine Learning



More Physical Example

- X → vector of Kinematic Variables
- Y → output : Higgs (Signal) or Background → 0 or 1
- Proposing a function like F(X)

ROOT

m
o
d
u
l
a
r
s
c
i
e
n
t
i
f
i
c
s
o
f
t
w
a
r
e
t
o
o
l
k
i
t

m
a
i
n
l
y
w
r
i
t
t
e
n
i
n
C
+
+

f
u
n
c
t
i
o
n
a
l
i
t
i

ROOT, Machine Learning → TMVA

- Toolkit for Multivariate Data Analysis
- Bunch of methods that provides a ROOT-integrated machine learning environment
- It includes Rectangular cut optimization, Boosted/Bagged decision trees, Artificial neural networks, …

TMVA :

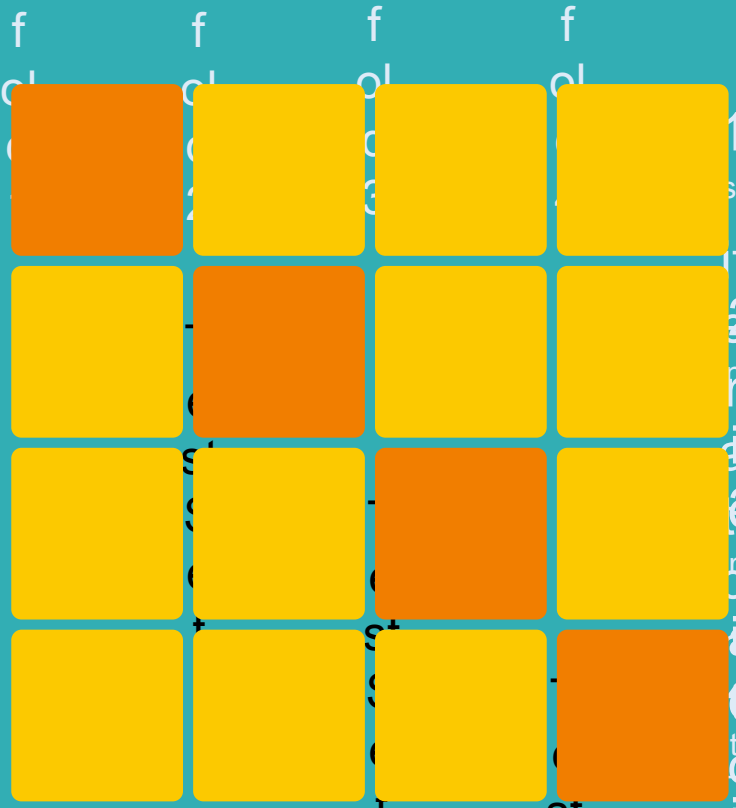
Training Phase → Training the method on known dataset

Application Phase → Applying the trained method to unknown dataset

Training Phase

- Create the Factory → The connection between the user and TMVA
- Giving the Training/Test trees
- Register input variables (Features)
- Select the MVA methods from the Factory that we are going to train on the data set
- Book, Train and Test the Mesthod

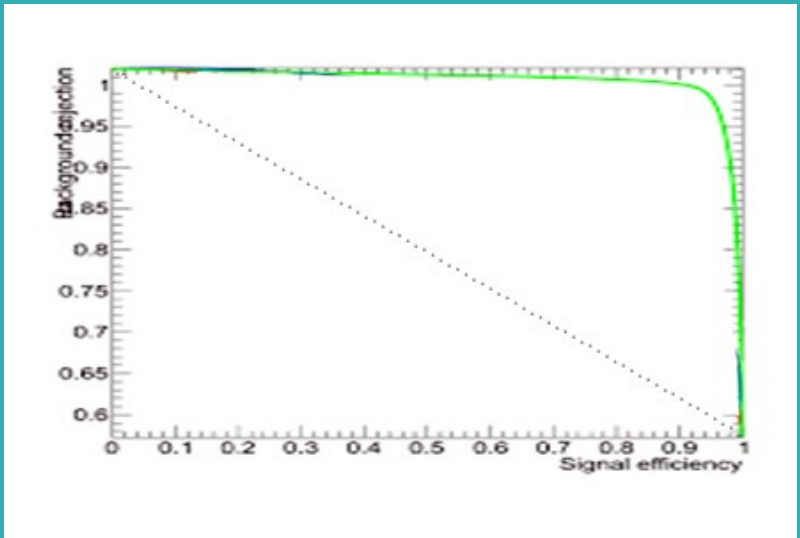
Cross Validation

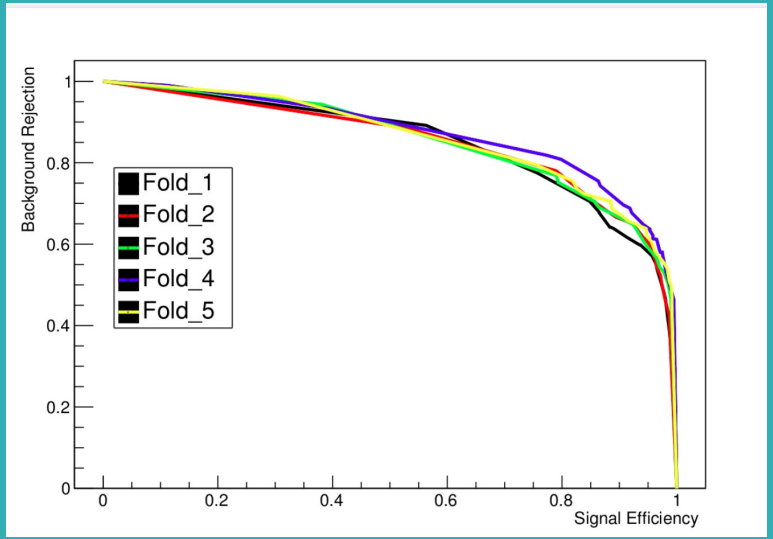


- Complete dataset
- Training dataset
- Test dataset

Calculate

Cross Validation: PlotROC()





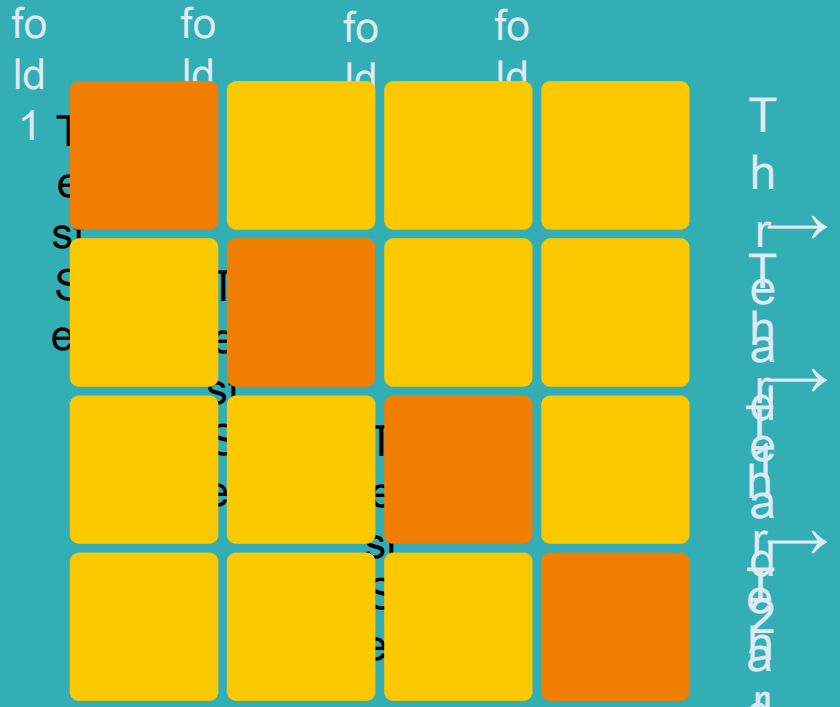
PlotROC()

ROC →Receiver Operating Characteristic: in Statistic graphical plot that illustrates the performance of a classifier

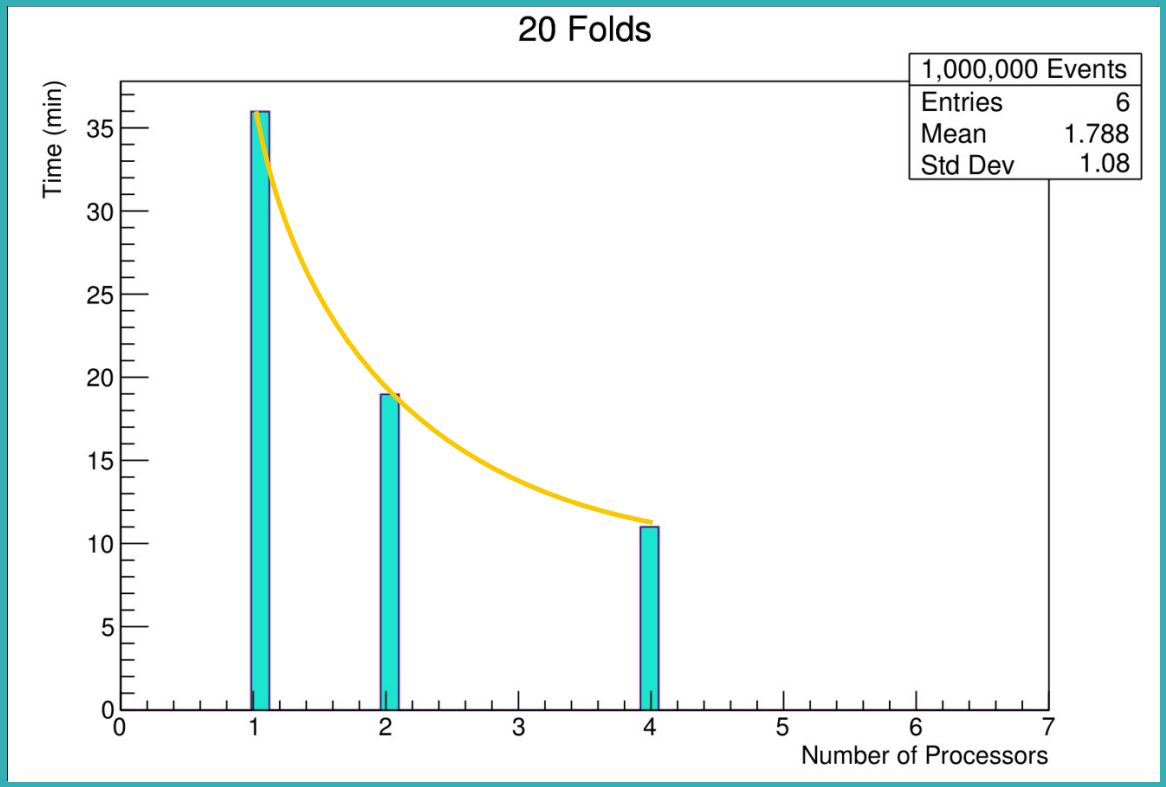
Parallelization

ROOT Classes for Parallelization

- **ThreadPool**
  →Multithreading
- **TProcPool**
  →Multiprocessing

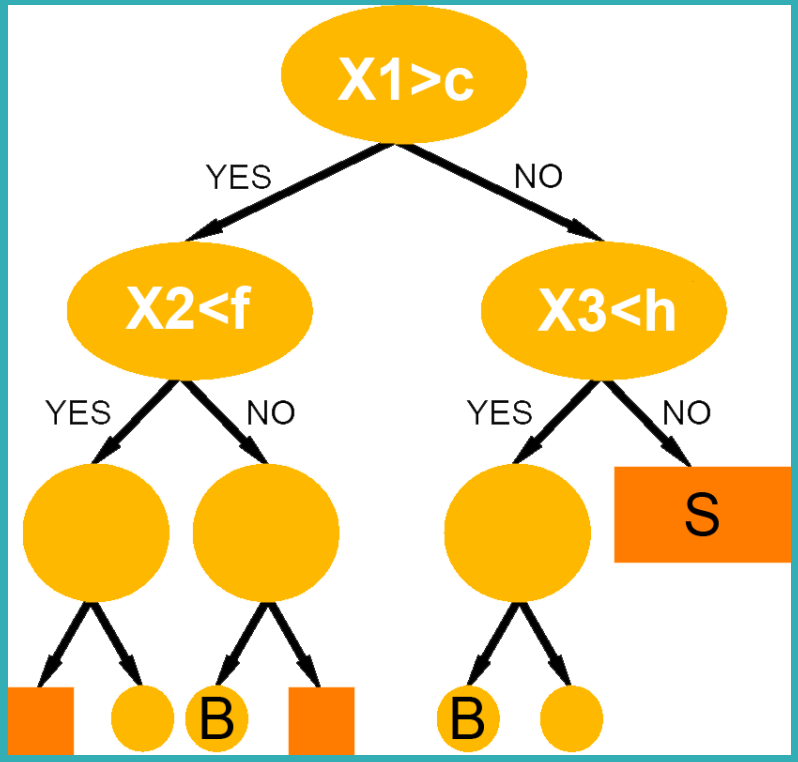


- **Multithreading**
  →More difficult to Implement : needs locking → no Global


20 Folds
1,000,000 Events
Entries 6
Mean 1.788
Std Dev 1.08
Time (min)
0
5
10
15
20
25
30
35
0
1
2
3
4
5
6
7
Number of Processors

Outlook

Parallelization of different methods like
BDT → Boosted Decision Tree



Conclusion

Thank you all very much for your attention!