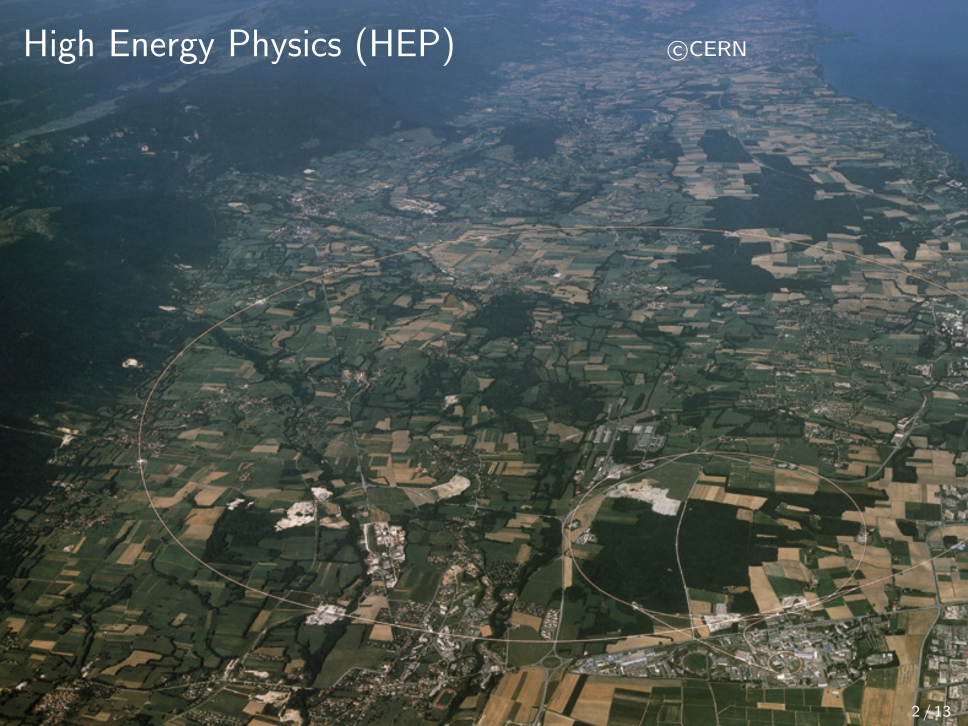
Scikit-Learn in particle physics

Gilles Louppe

CERN, Switzerland

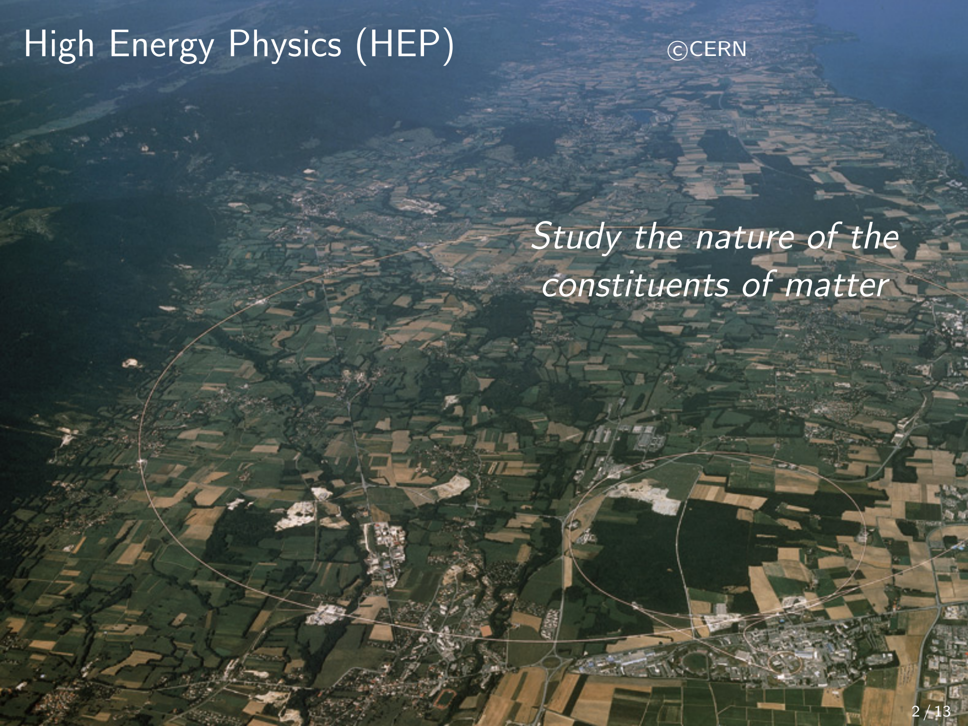
November 18, 2014

High Energy Physics (HEP)



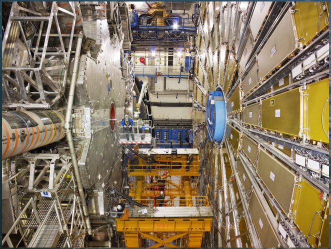
©CERN

High Energy Physics (HEP)

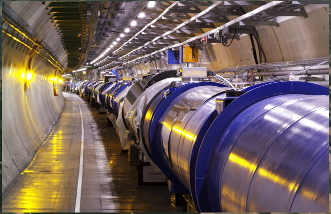


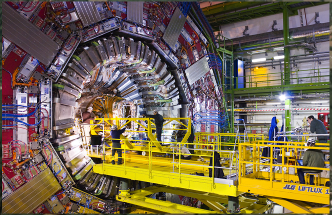
Study the nature of the constituents of matter

High Energy Physics (HEP)



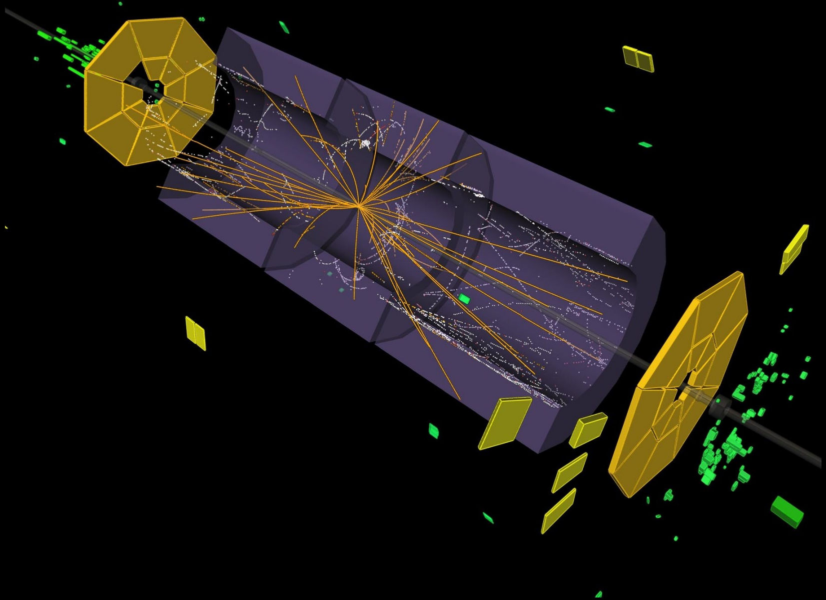
Study the nature of the constituents of matter





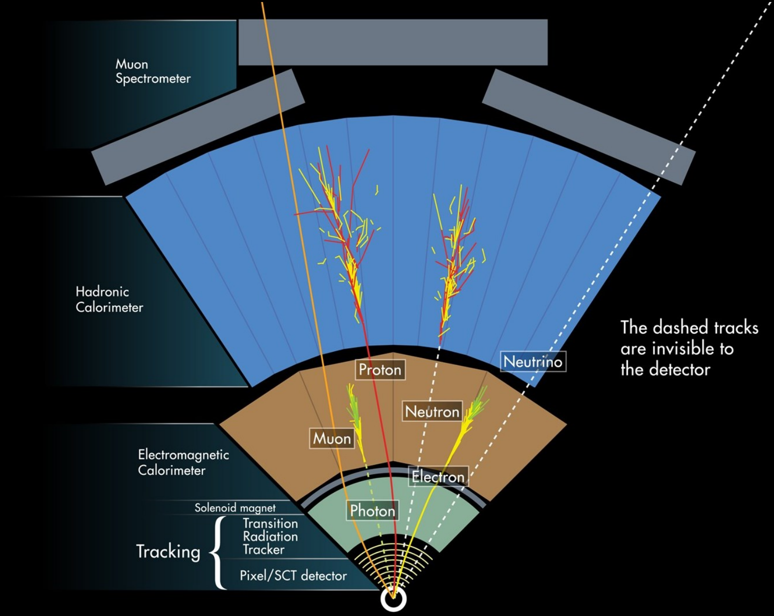
Particle detector 101

©ATLAS CERN



Particle detector 101

©ATLAS CERN



Data analysis tasks in detectors

1. Track finding

   Reconstruction of particle trajectories from hits in detectors

2. Budgeted classification

   Real-time classification of events in triggers

3. Classification of signal / background events

   Offline statistical analysis for discovery of new particles

The Kaggle Higgs Boson challenge (in HEP terms)

- Data comes as a finite set

$$\mathcal{D} = \{(\mathbf{x}_i, y_i, w_i) | i = 0, \ldots, N-1\},$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{\text{signal}, \text{background}\}$ and $w_i \in \mathbb{R}^+$.

- The goal is to find a region $\mathcal{G} = \{\mathbf{x} | g(\mathbf{x}) = \text{signal}\} \subset \mathbb{R}^d$, defined from a binary function g, for which the background-only hypothesis can be rejected at a strong significance level ($p = 2.87 \times 10^{-7}$, i.e., *5 sigma*).

- Empirically, this is approximately equivalent to finding g from $\mathcal{D}$ so as to maximize AMS $\approx \frac{s}{\sqrt{b}}$, where
  - $s = \sum_{\{i | y_i = \text{signal}, g(\mathbf{x_i}) = \text{signal}\}} w_i$
  - $b = \sum_{\{i | y_i = \text{background}, g(\mathbf{x_i}) = \text{signal}\}} w_i$

The Kaggle Higgs Boson challenge (in ML terms)

Find a binary classifier

$$g : \mathbb{R}^d \mapsto \{\text{signal}, \text{background}\}$$

maximizing the objective function

$$AMS \approx \frac{s}{\sqrt{b}},$$

where

- s is the weighted number of true positives
- b is the weighted number of false positives.

Winning methods

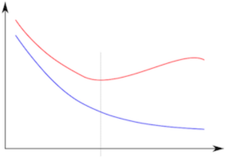
#	Δ1w	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑4	Gábor Melis ‡ *	3.80581	110	Sun, 14 Sep 2014 09:10:04 (-0h)
2	↓1	Tim Salimans ‡ *	3.78913	57	Mon, 15 Sep 2014 23:49:02 (-40.6d)
3	—	nhlx5haze ‡ *	3.78682	254	Mon, 15 Sep 2014 16:50:01 (-76.3d)
4	↑55	ChoKo Team	3.77526	216	Mon, 15 Sep 2014 15:21:36 (-42.1h)
5	↑23	cheng chen	3.77384	21	Mon, 15 Sep 2014 23:29:29 (-0h)

- Ensembles of neural networks (1st and 3rd) ;
- Ensembles of regularized greedy forests (2nd) ;
- Boosting with regularization (XGBoost package).

- Most contestants dit not optimize AMS directly ;
- But chosed the prediction cut-off maximizing AMS in CV.

Lessons learned (for machine learning)

- AMS is highly unstable, hence the need for
  - Rigorous and stable cross-validation to avoid overfitting.
  - Ensembles to reduce variance ;
  - Regularized base models.



- Support of samples weights w_i in classification models was key for this challenge.

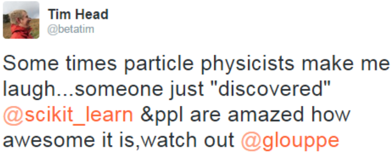
- Feature engineering hardly helped.

Lessons learned (for physicists)

- Domain knowledge hardly helped.
- Standard machine learning techniques, run on a single laptop, beat benchmarks without much efforts.
- Physicists started to realize that collaborations with machine learning experts is likely to be beneficial.
  - *I worked on the ATLAS experiment for over a decade [...] It is rather depressing to see how badly I scored.*
  - *The final results seem to reinforce the idea that the machine learning experience is vastly more important in a similar contest than the knowledge of particle physics. I think that many people underestimate the computers.*
  - *It is probably the reason why ML experts and physicists should work together for finding the Higgs.*

Scientific software in HEP

- ROOT and TMVA are standard data analysis tools in HEP.
- Surprisingly, this HEP software ecosystem proved to be rather limited and easily outperformed (at least in the context of the Kaggle challenge).
- Yet, the adoption of external solutions (e.g., the scientific Python stack) appears to be slow and difficult because of
  - No major added-value ;
  - The learning curve of new tools ;
  - Lack of understanding of non-HEP methods ;
  - Isolation from the community ;
  - Genuine ignorance.



Scikit-Learn in Particle Physics ?

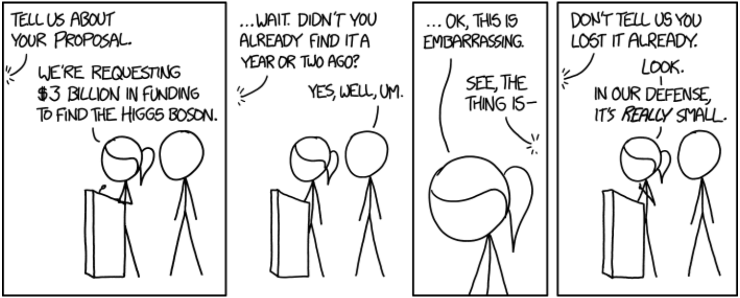
- The main technical blocker for the larger adoption of Scikit-Learn in HEP remains the full support of sample weights throughout all essential modules.
  - Since 0.16, weights are supported in all ensembles and in most metrics.
  - Next step is to add support in grid search.

- In parallel, domain-specific packages are getting traction
  - `ROOTpy`, for bridging the gap between ROOT data format and NumPy ;
  - `lhcb_trigger_ml`, implementing ML algorithms for HEP (mostly Boosting variants), on top of scikit-learn.
- Major blocker : political reasons ?

Conclusions

- Scikit-Learn has the potential to become an important tool in HEP. But we are not there yet [WIP].
- Overall, both for data analysis and software aspects, this calls for a larger collaboration between data sciences and HEP.

The process of attempting as a physicist to compete against ML experts has given us a new respect for a field that (through ignorance) none of us held in as high esteem as we do now.



©xkcd

Questions ?