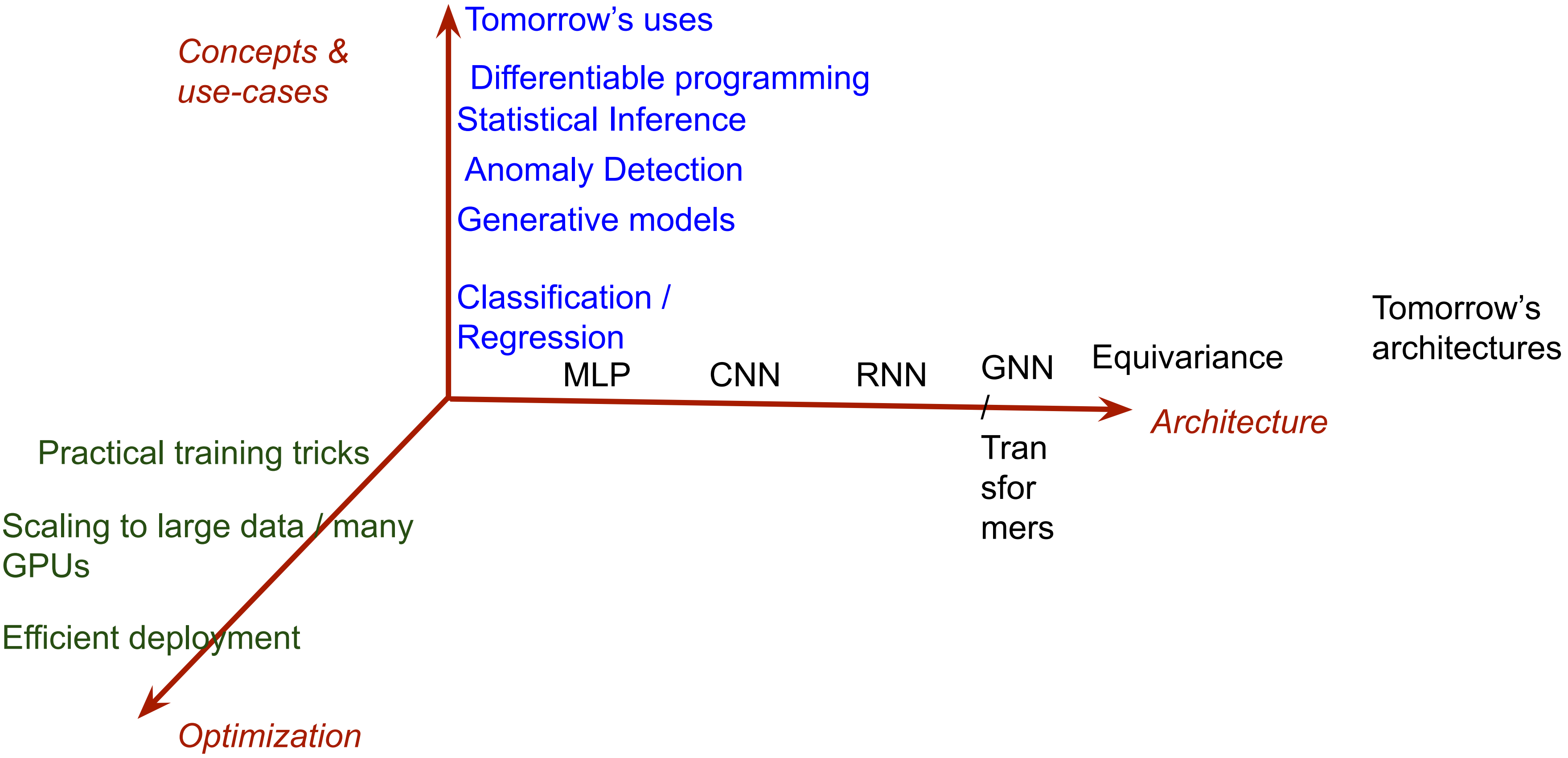
The bread and butter ML in HEP

Aishik Ghosh and **Elham E Khoda**

ML4HEP ICTS
31 August, 2023

ML for HEP


Concepts &
use-cases
Tomorrow's uses
Differentiable programming
Statistical Inference
Anomaly Detection
Generative models
Classification /
Regression
MLP
CNN
RNN
GNN / Transformers
Equivariance
Tomorrow's
architectures
Architecture
Practical training tricks
Scaling to large data / many GPUs
Efficient deployment
Optimization

The plan

Today: Typical ML for signal vs background classification in HEP analysis, not trivial when you think about the gory details

Tomorrow: Neural simulation-based inference (SBI), which is ML for statistics (parameter inference, unfolding, uncertainties)

Monday: More on SBI, Generative models

Thu, 31, Aug	Tommaso Dorigo	**Tea Break**	Sanmay Ganguly	**Lunch Break**	Elham E Khoda, Aishik Ghosh	**Tea Break**	Elham E Khoda, Aishik Ghosh	(Colloquium) Jan Kieseler*
Fri, 01, Sep	Tommaso Dorigo	**Tea Break**	Sanmay Ganguly	**Lunch Break**	Elham E Khoda, Aishik Ghosh	**Tea Break**	Elham E Khoda, Aishik Ghosh	(Colloquium) Jia Liu*
Mon, 04, Sep	Elham E Khoda, Aishik Ghosh	**Tea Break**	Elham E Khoda, Aishik Ghosh	**Lunch Break**	Elham E Khoda, Aishik Ghosh	**Tea Break**	Elham E Khoda, Aishik Ghosh	(Colloquium) Michael Kagan*

All concepts will be complementary to what you have learnt about architectures (MLPs, CNNs, GNNs, transformers). Tutorials with simplified data and architectures to focus on concepts and allow fast training

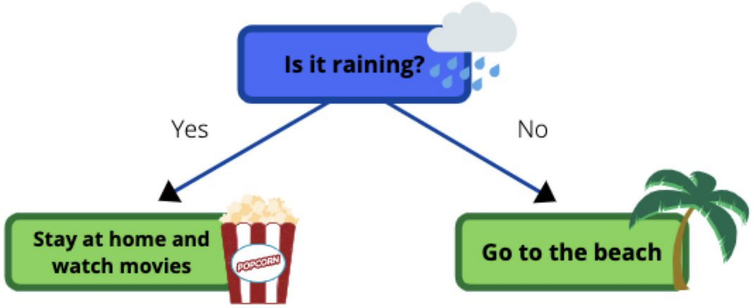
Decision Trees

Decision trees are simple programs consisting of a nested sequence of “if-else” decisions based on the features (splitting rules).

- Decision tree algorithm falls under the category of supervised learning
- They can be used to solve both **regression** and **classification** problems

An overly simplified example:

Do I go to the beach or do I stay at home and watch movies?
Well, the answer almost always depends on what the weather is like on the day



Structure of a Decision Tree

Tree -like graphs

Nodes: A place where we pick an attribute and ask a question

- Data is split at each node

Leaves: Terminal nodes

- Represent a class label or probability
- Continuous outcome: "regression tree"

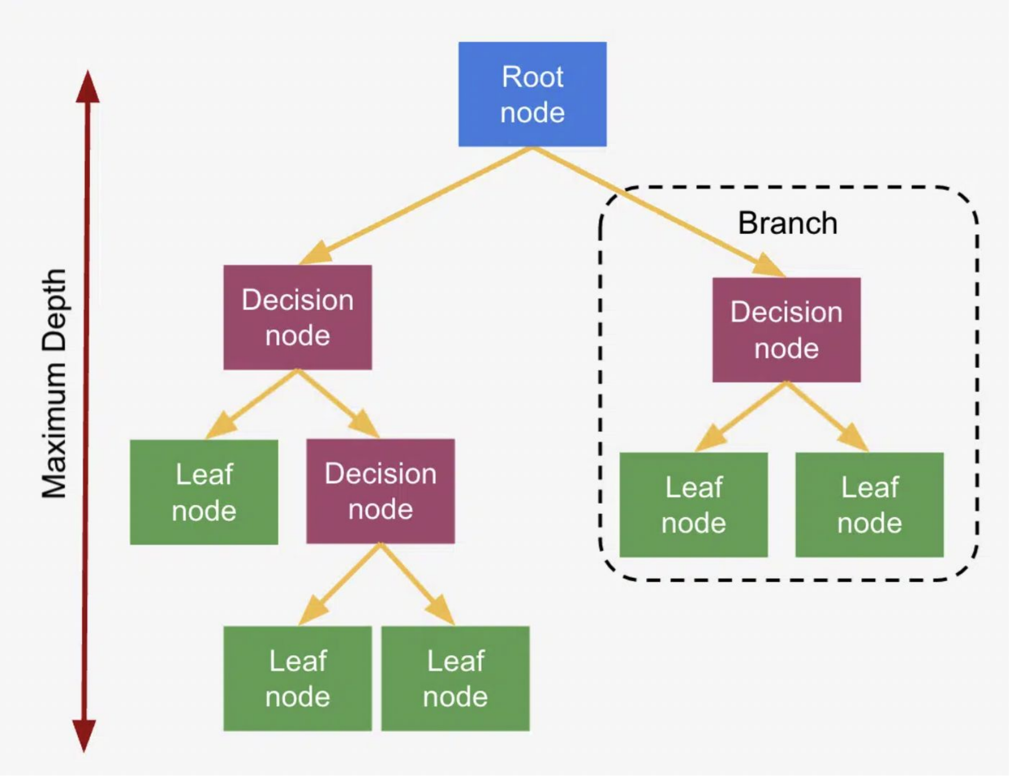


Image source

Decision Tree: Learning

A decision tree takes a set of input features and splits input data recursively based on the conditions on those features.

How to choose the root node?

- Entropy/ Gini impurity
- Information gain

Entropy and Gini impurity measures the purity of split

- Lower the entropy/Gini impurity → the better

Gini / Entropy = 0

When all training instances belong to the same class

Entropy Formula

$$\text{Entropy} = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

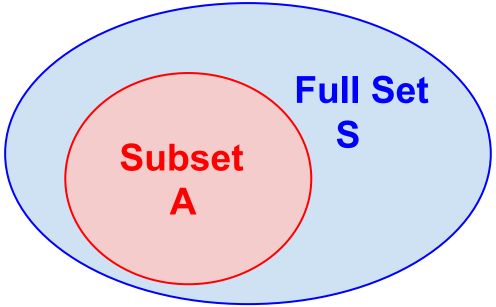
Gini Impurity Formula

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

n = total number of classes

p_i is the probability of a certain classification i

Decision Tree: Learning



Subset

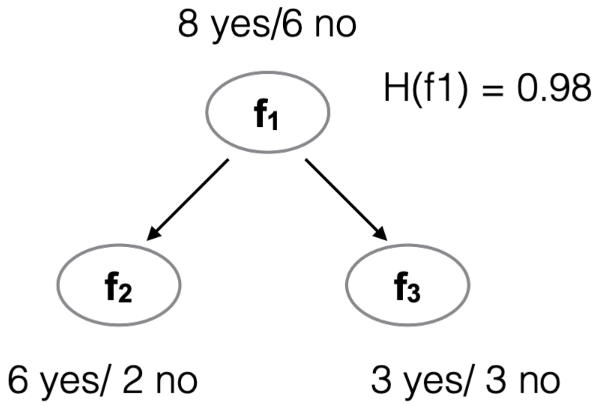
$$\text{Gain}\ (S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

Subset Entropy

Full Set

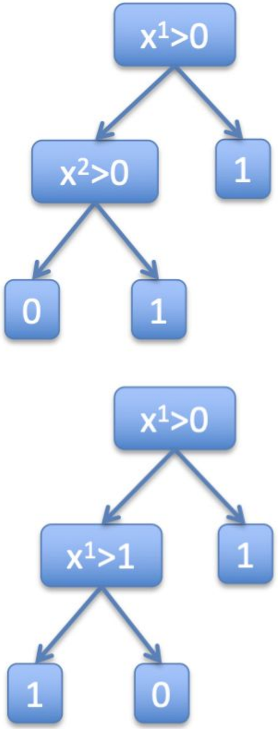
Total Entropy

$$\begin{aligned}\text{Gain}(S, f_1) &= H(S) - \sum_{v \in \text{values}(f_1)} \frac{|S_v|}{|S|} H(S_v) \\ &= H(S) - \frac{8}{14} H(f_2) - \frac{6}{14} H(f_3) \\ &= 0.98 - \frac{8}{14} \times 0.81 - \frac{6}{14} \times 1\end{aligned}$$



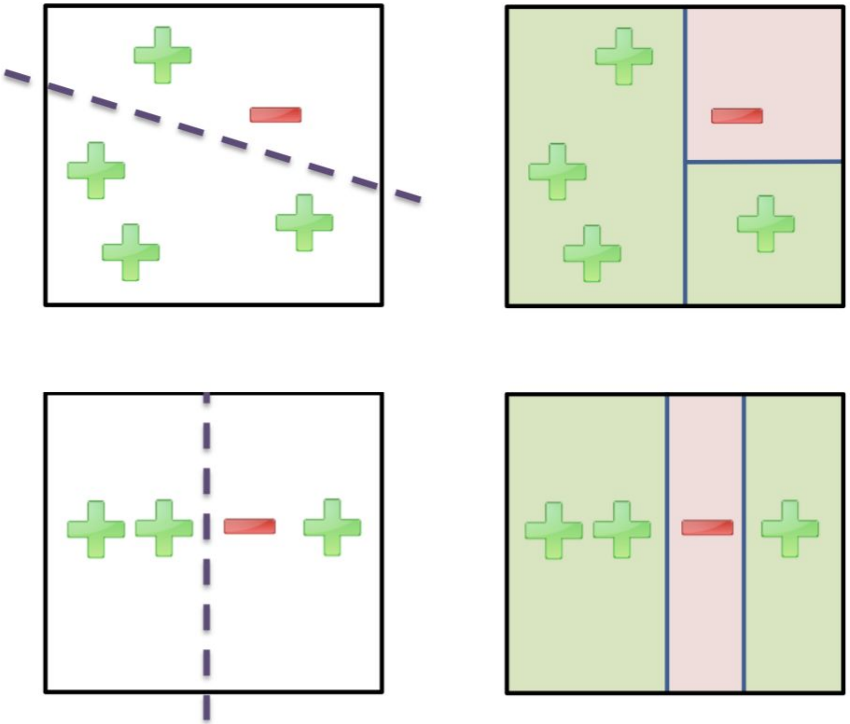
$$H(S) = -\frac{8}{14} \times \log_2(\frac{8}{14}) - \frac{6}{14} \times \log_2(\frac{6}{14})$$

Decision Trees are nonlinear models



No linear model can achieve 0 error

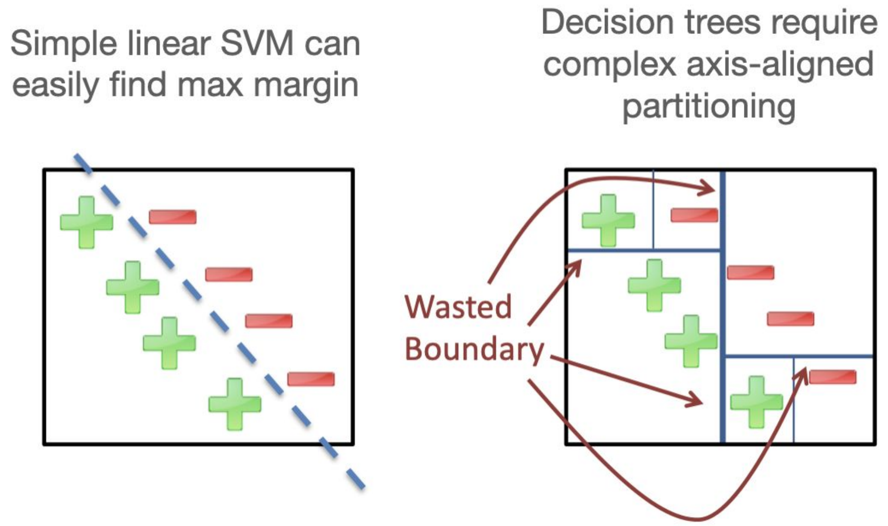
Simple decision tree can achieve 0 error



Decision Trees are Axis-aligned

Decision Trees are axis-aligned

- *Cannot easily model diagonal boundaries*

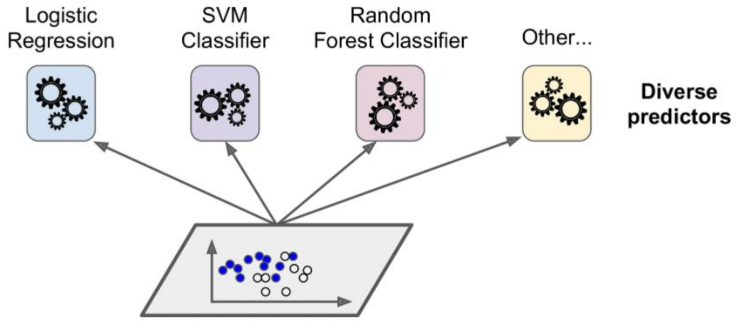


Ensemble Techniques

Aggregate the predictions of a group of predictors (ex. classifier)

→ **Often get better predictions than with the best individual predictor**

A group of predictors is called an ensemble



- **Simple Techniques:**
  - Max voting, Averaging, Weighted Averaging
- **Advanced Techniques:**
  - Stacking, Blending, Bagging, **Boosting**

Link to an animation: Random Forest

Image: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Boosting (the "B" of BDT)

Boosting combines several weak learners into a strong learner

- **Sequential process** → train predictors sequentially, each trying to correct its predecessor

Some popular boosting algorithms:

- Adaptive Boosting (AdaBoost)
- Gradient Boosting (GBM)
- Extreme Gradient Boosting (XGBoost)
- Light Gradient Boosting (LGBM)

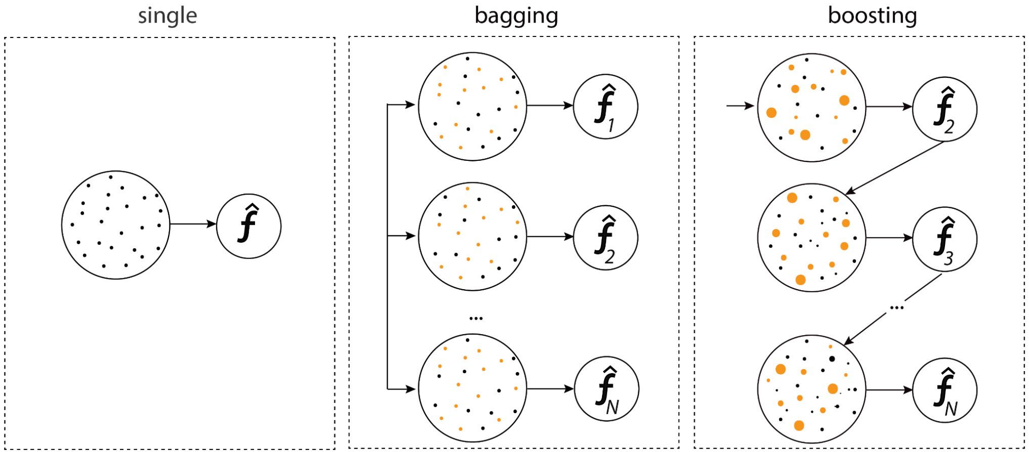


Image source

Gradient Boosting

We are building a weighted sum of weak learners

"Boosting" or improving a single weak model by combining it with a number of other weak models

→ generate a collectively strong model

→ *Reduces bias of weak learners*

- Iteratively train an ensemble of shallow decision trees
- With each iteration using the error residuals of the previous model to fit the next model
- The final prediction is a weighted sum of all of the tree predictions

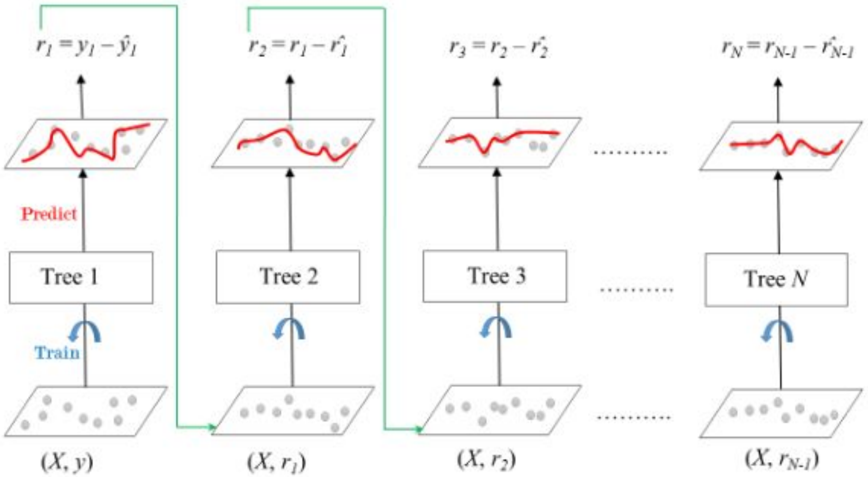


Image source

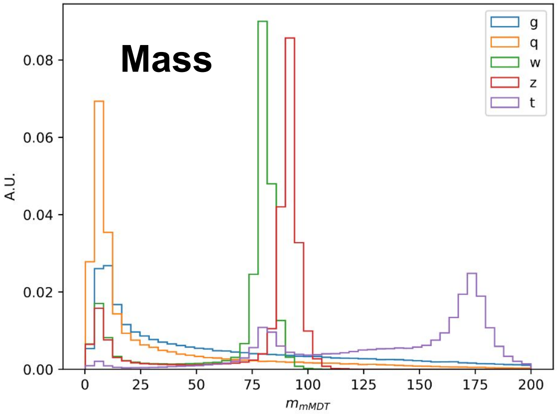
BDTs are efficient for tabular data

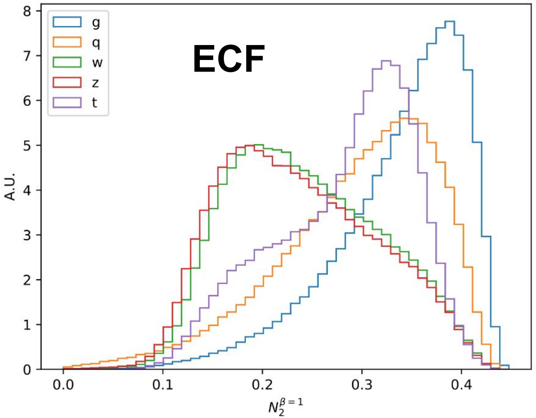
Tabular data: Use physics use physics knowledge to preprocess event information into a set of high-level features

BDTs are still effective for **tabular data**

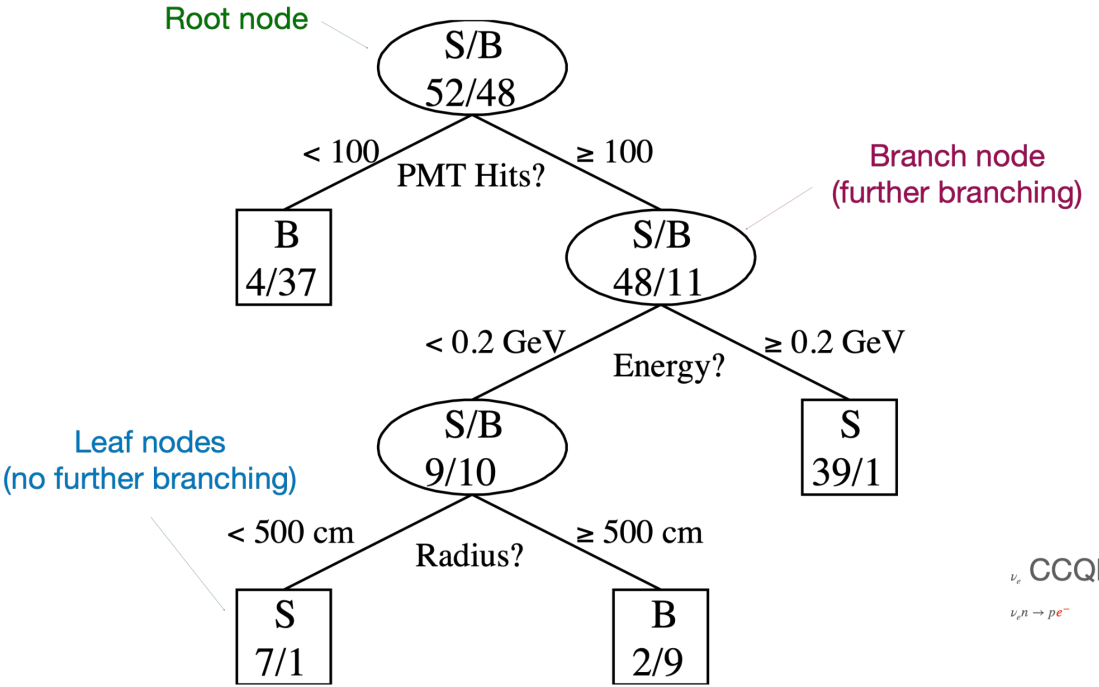
Example: Jet classification

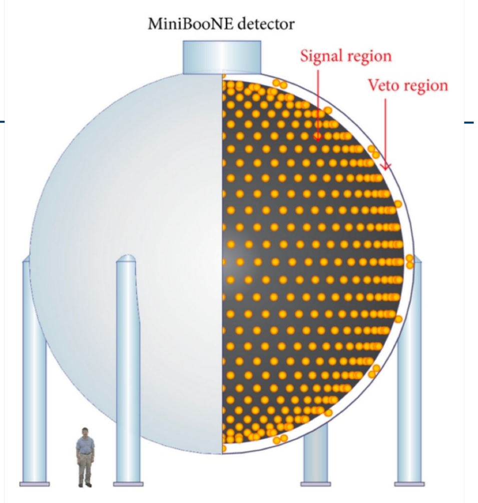
- Substructure variable, jet mass, energy correlation functior $N_2^{\beta=1} = {}_2e_3^{\beta=1}/({}_1e_3^{\beta=1})^2$



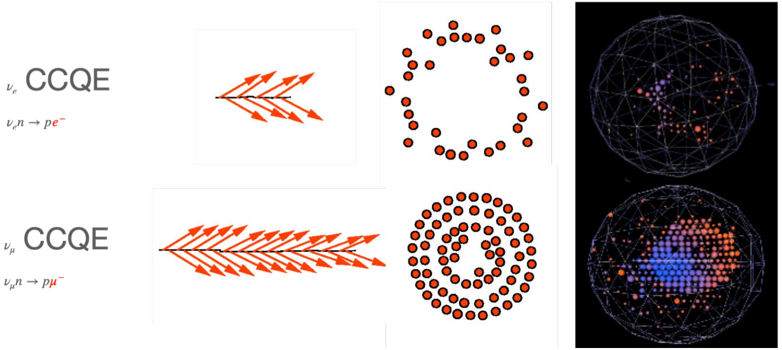


Application: MiniBooNE





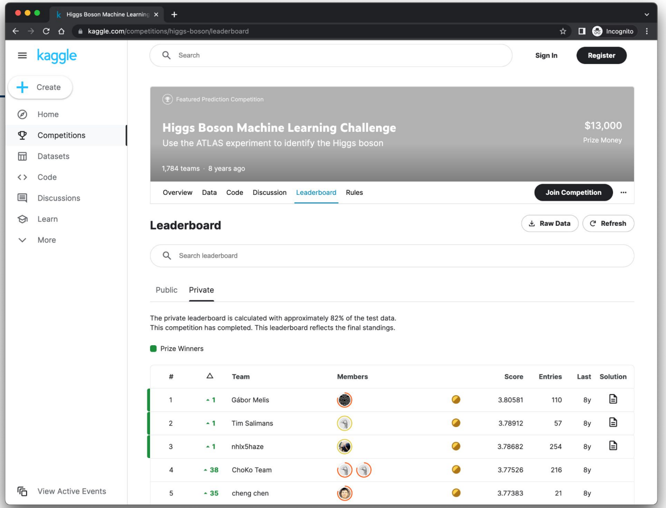
MiniBooNE: 1520 photomultiplier signals
Goal: separate ν_e and ν_μ events

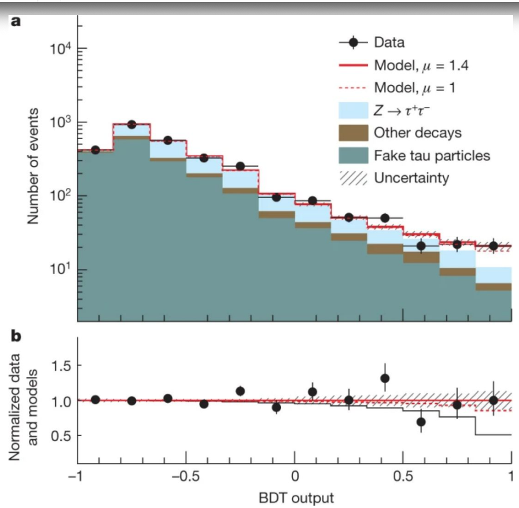


slide from Javier Duarte

BDTs in the Wild

- One of the winners of Kaggle Higgs Boson Machine Learning Challenge [kaggle.com/competitions/higgs-boson]
  - And many other uses at LHC, e.g. in Higgs boson discovery [10.1038/s41586-018-0361-2]
- Predicting critical temperature of a superconductor [10.1016/j.commatsci.2018.07.052]
- MiniBooNE neutrino event classification [10.1016/j.nima.2004.12.018]
- Observation of single top quark production at D0 [10.1103/PhysRevLett.103.092001]





Common BDT Packages

XGBoost

- Shot to fame as one of the winners of the HiggsML challenge
- Engineered for speed, parallelisation, includes regularization tricks
- Cannot handle negative weighted events

LightGBM

- Open source, backed by Microsoft
- Even faster, scales to massive datasets, inbuilt clever pre-processing
- Can handle negative weights, categorical variables (Eg. “ggF region”, “VBF region”)

CatBoost

- Yandex backed
- Best for categorical variables

(TMVA not recommended. Yes, even for HEP data. Yes, even if data is in .root files)

BDT Hyperparameters

- Learning rate
- Bin size for histogramming
- Treatment of categorical variables
- Bagging faction & feature fraction
- Min events per leaf
- Number of estimators / trees
- Max Depth
- Pruning
- .....

Guide to HPO for LightGBM:
https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

For Better Accuracy

- Use large `max_bin` (may be slower)
- Use small `learning_rate` with large `num_iterations`
- Use large `num_leaves` (may cause over-fitting)
- Use bigger training data
- Try `dart`

Deal with Over-fitting

- Use small `max_bin`
- Use small `num_leaves`
- Use `min_data_in_leaf` and `min_sum_hessian_in_leaf`
- Use bagging by set `bagging_fraction` and `bagging_freq`
- Use feature sub-sampling by set `feature_fraction`
- Use bigger training data
- Try `lambda_l1`, `lambda_l2` and `min_gain_to_split` for regularization
- Try `max_depth` to avoid growing deep tree
- Try `extra_trees`
- Try increasing `path_smooth`

‘Permutation importance’ - A more relevant metric

Take a **trained classifier** and test its response on a test set by **shuffling the values (between events) of one feature (variable) at a time**.

Retain the marginal distribution but break all correlations with the events

See how the performance deteriorates.

- Define your own performance metric, such as significance (Z)
- Works also for Neural Networks

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...	...	...	...
156	142	...	8
153	130	...	24

Warning: All feature importance methods have certain weaknesses, like when variables are correlated

Let’s practice!

We will train a classifier to discriminate Higgs ($\rightarrow WW$) signals from other Standard Model backgrounds

There are two notebooks:

- Learn the concepts with the BDT notebook: https://github.com/ml4hep-India/icts-2023/blob/main/higgs_classification/HEPML_HandsOn_BDT.ipynb
- More interactive for the NN notebook: https://github.com/ml4hep-India/icts-2023/blob/main/higgs_classification/HEPML_HandsOn_NN.ipynb

Some Notes

- Decision Trees require very little data preparation: feature scaling or centering is not required
- Scikit-Learn uses the CART algorithm, which produces only binary trees
- other algorithms such as ID3 can produce Decision Trees with nodes that have more than two children