# Machine Learning

Prof. Sergei Gleyzer

Lecture 7

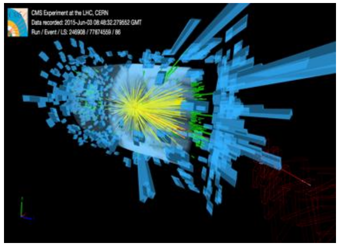
**PH451, PH551**

**February 11, 2025**

# Announcements

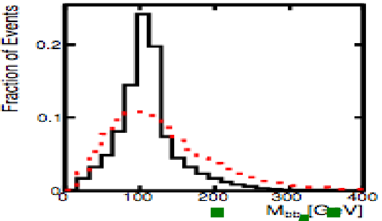
- **Mini-Hackathon #1**
  - **due Fri, Feb. 21 at 5pm**
- **This week: HS #4**
  - **due next Tue. 1pm**

# Higgs Boson Challenge


CMS Experiment at the LHC, CERN
Data recorded: 2015-Jun-03 08:48:32.279552 GMT
Run / Event / LS: 246908 / 77874559 / 86

## Dataset:

- https://archive.ics.uci.edu/ml/datasets/HIGGS



## Paper with detailed description

- https://arxiv.org/pdf/1402.4735.pdf
- Classify Higgs Boson signal from similar-looking background

# Recap: Ensemble Methods

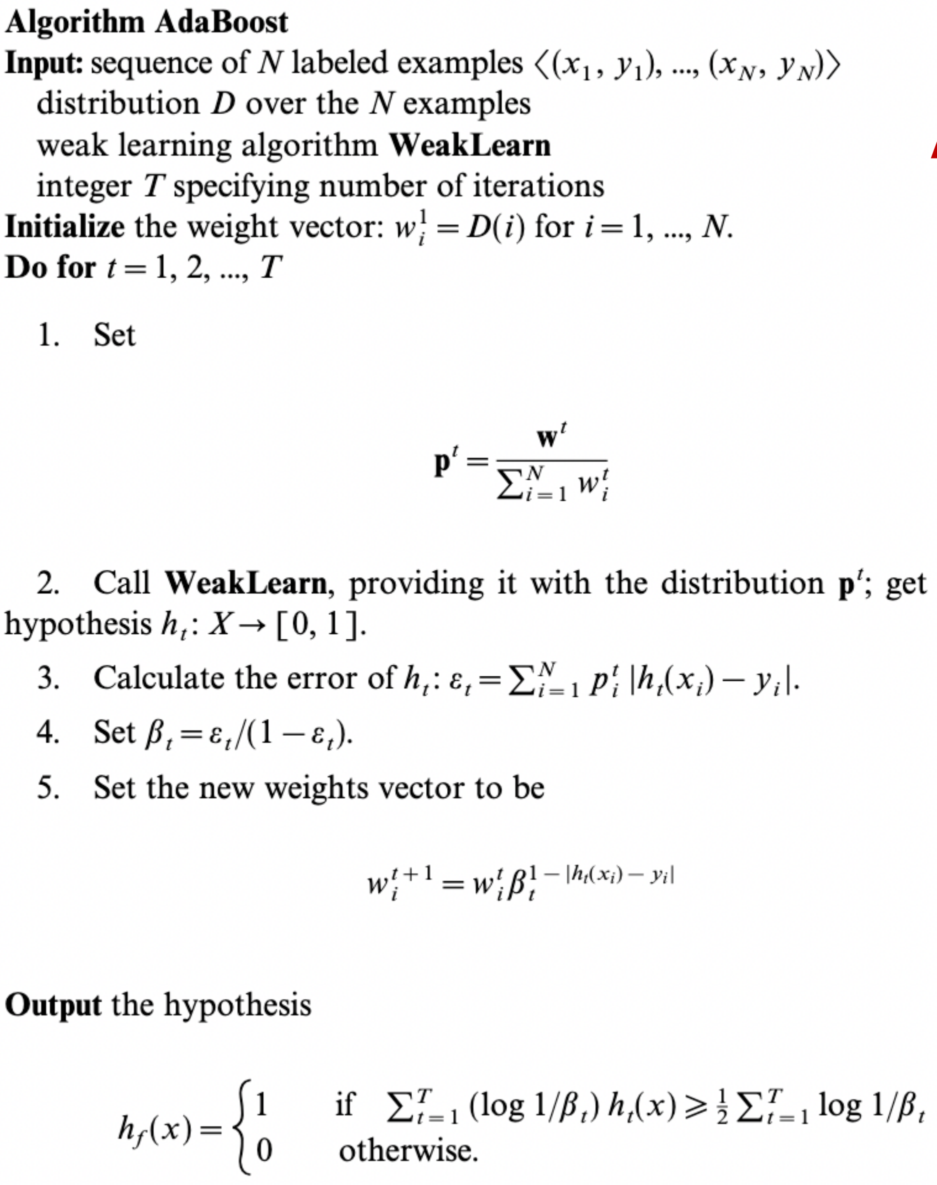
Suppose you have a **collection** of discriminants $f\,(x, w_k)$, which, individually, perform only **marginally** better than random guessing.

$$f\,(x) = a_0 + \sum_{k=1}^{K} a_k\, f\,(x, w_k)$$

From such discriminants, **weak learners**, it is possible to build highly effective ones by averaging over them:

Friedman and Popescu (2008) DOI:10.1214/07-AOAS148

# AdaBoost



```
Algorithm AdaBoost
Input: sequence of $N$ labeled examples $\langle (x_1, y_1), ..., (x_N, y_N) \rangle$
    distribution $D$ over the $N$ examples
    weak learning algorithm WeakLearn
    integer $T$ specifying number of iterations
Initialize the weight vector: $w_i^1 = D(i)$ for $i = 1, ..., N$.
Do for $t = 1, 2, ..., T$
    1. Set
           $\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^N w_i^t}$
    2. Call WeakLearn, providing it with the distribution $\mathbf{p}^t$; get back a
       hypothesis $h_t \colon X \to [0, 1]$.
    3. Calculate the error of $h_t$: $\varepsilon_t = \sum_{i=1}^N p_i^t \, |h_t(x_i) - y_i|$.
    4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.
    5. Set the new weights vector to be
           $w_i^{t+1} = w_i^t \beta_t^{1 - |h_t(x_i) - y_i|}$
Output the hypothesis
    $h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T (\log 1/\beta_t)\, h_t(x) \geqslant \frac{1}{2} \sum_{t=1}^T \log 1/\beta_t \\ 0 & \text{otherwise.} \end{cases}$
```

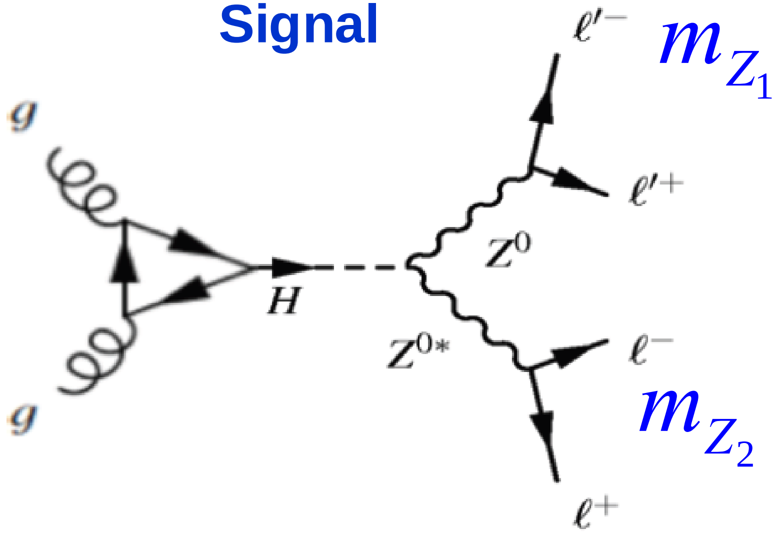
**FIG. 2.** The adaptive boosting algorithm.

**Y. Freund and Schapire (1997)**

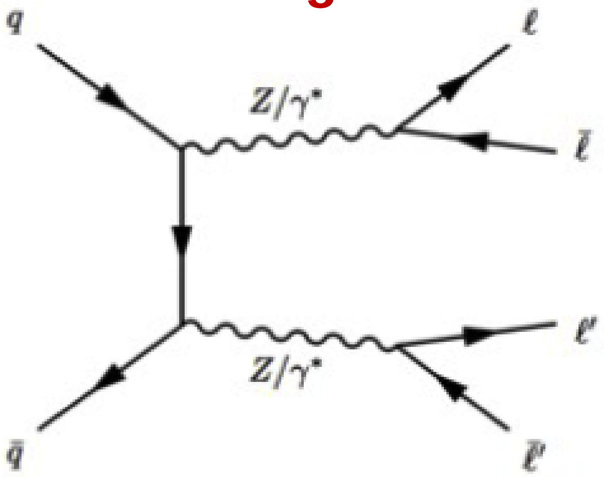
# Illustrative Example

# H → ZZ* → 4 leptons

Signal



Background



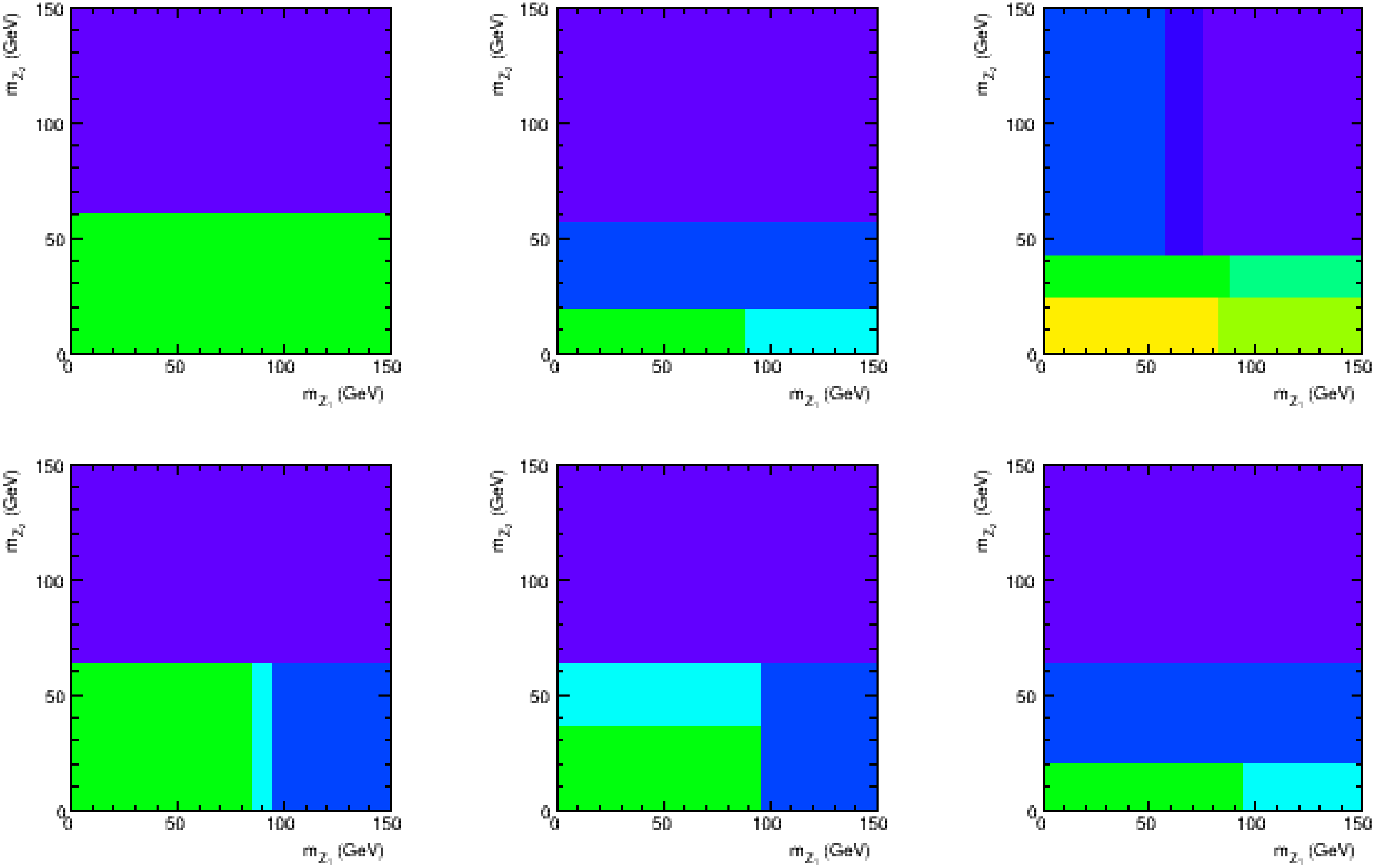
$pp ® H ® ZZ ® \ell^+\ell^- \ell¢^+\ell¢$

$pp ® ZZ ® \ell^+\ell^- \ell¢^+\ell¢$

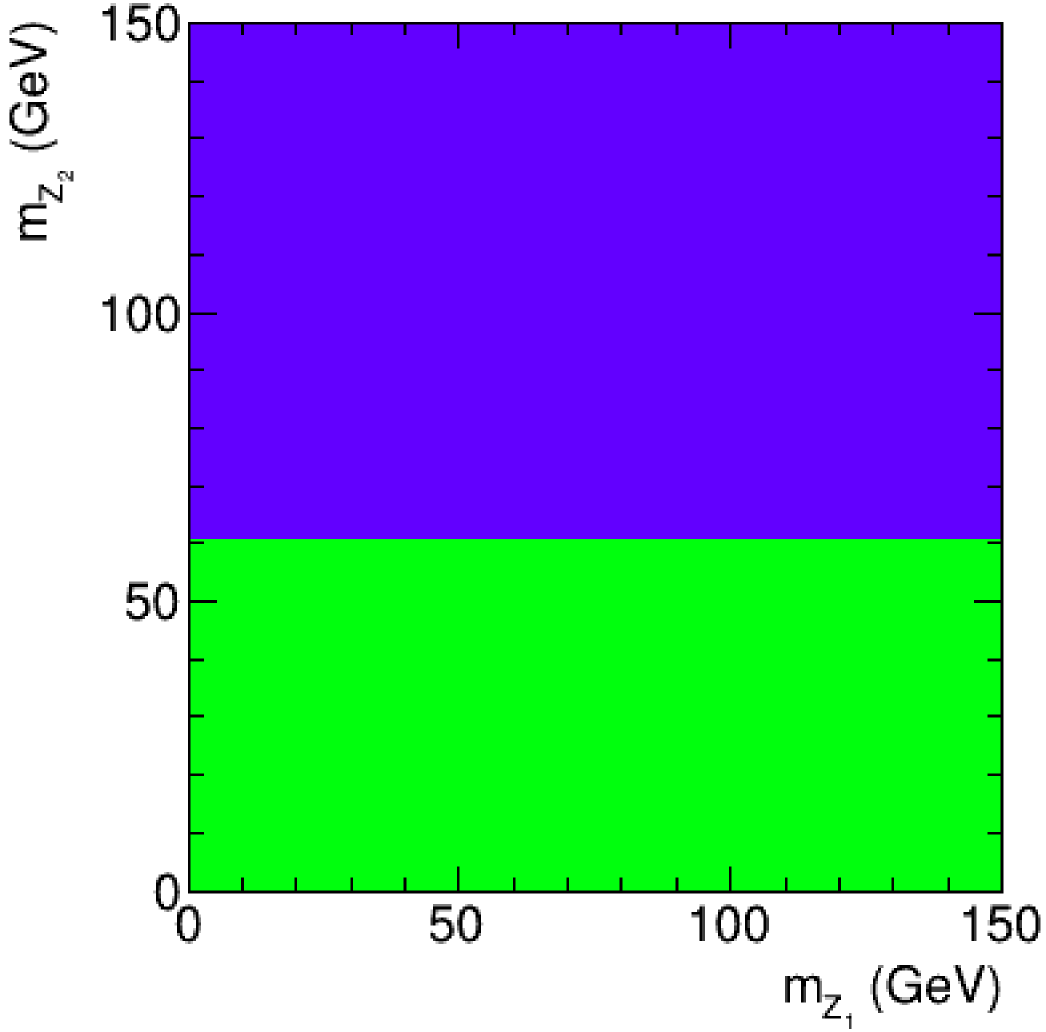
**$x = (m_{Z1}, m_{Z2})$**

Credit: H. Prosper

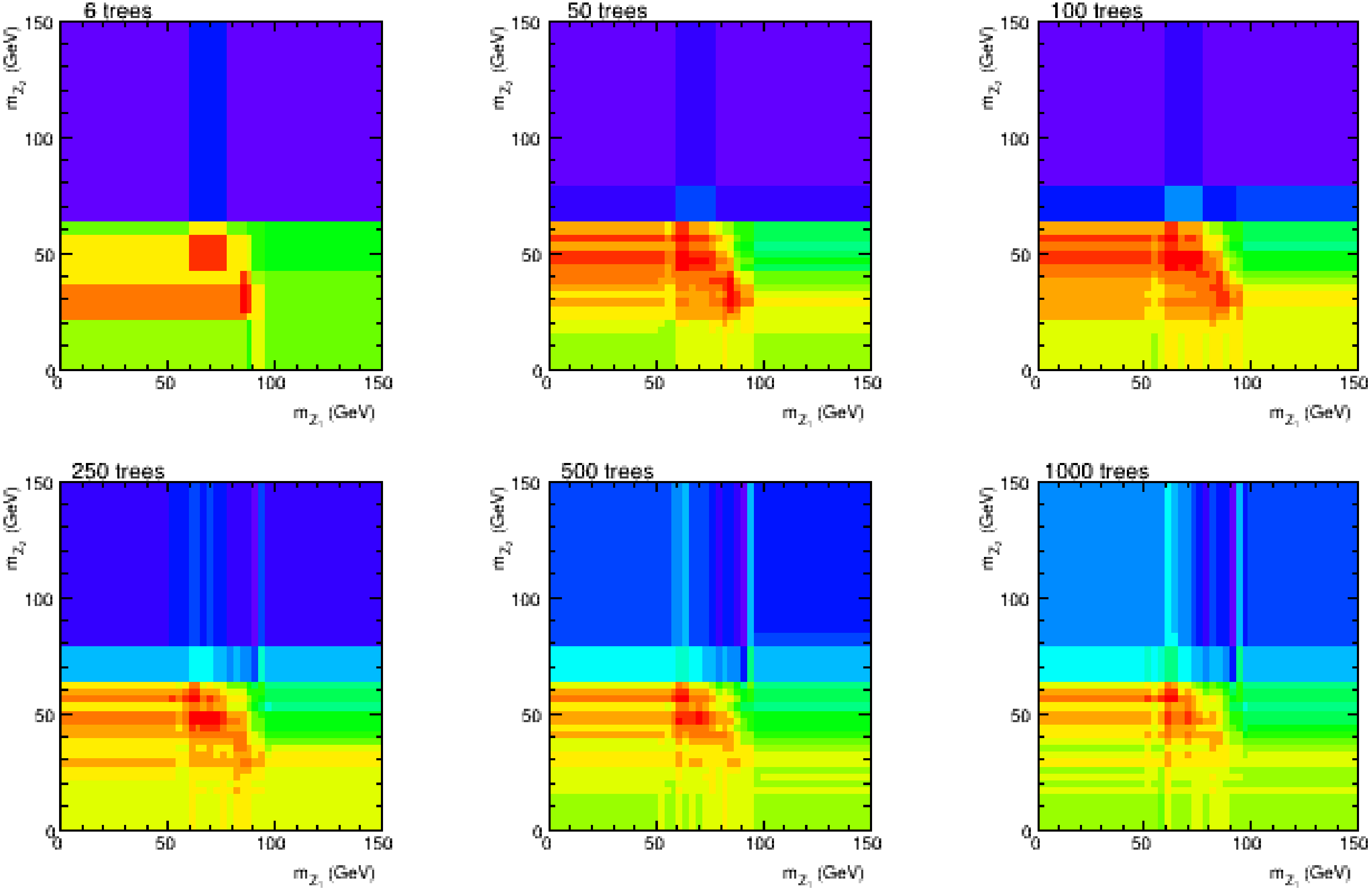
# First 6 Decision Trees



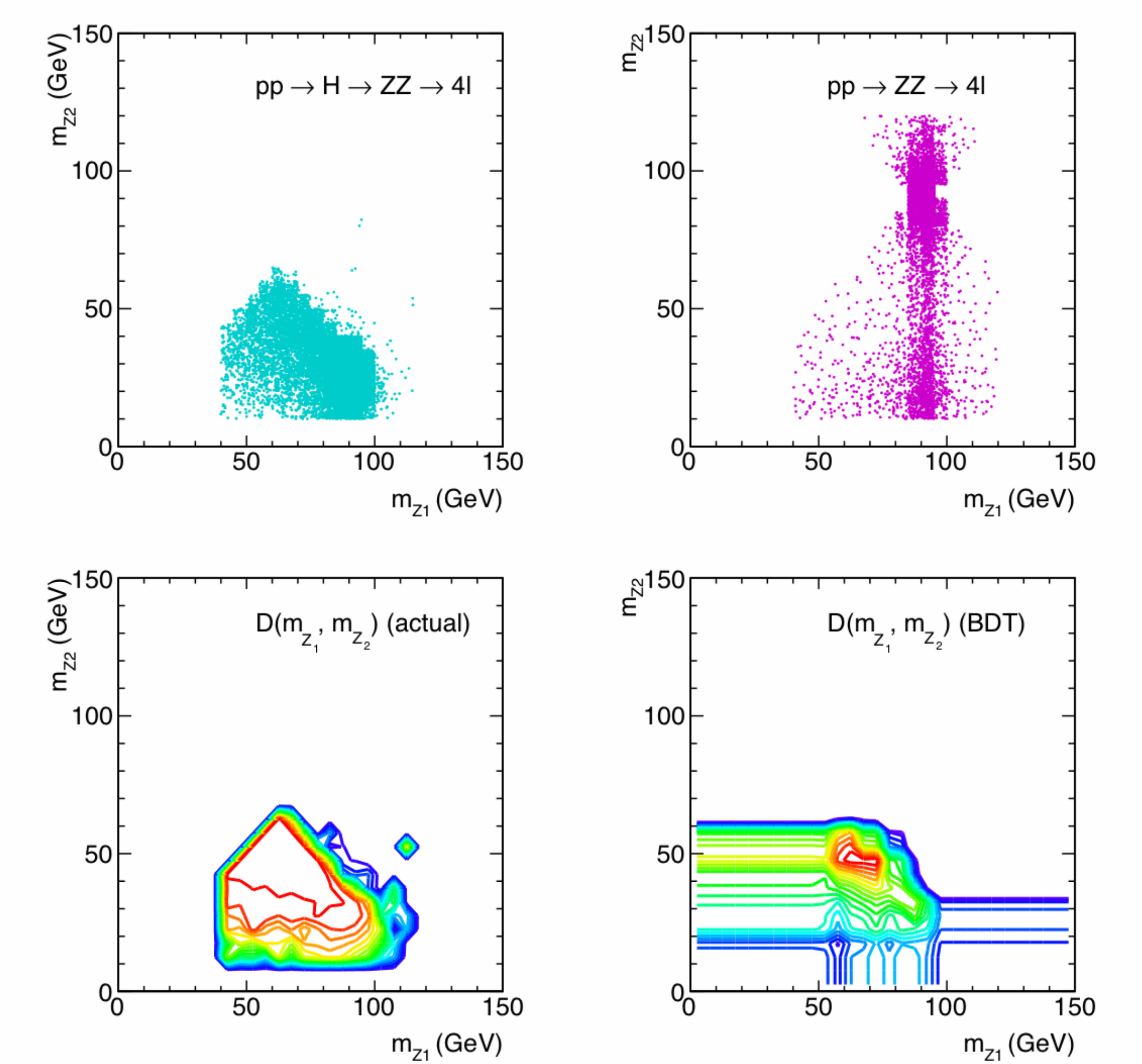
# First 100 Decision Trees



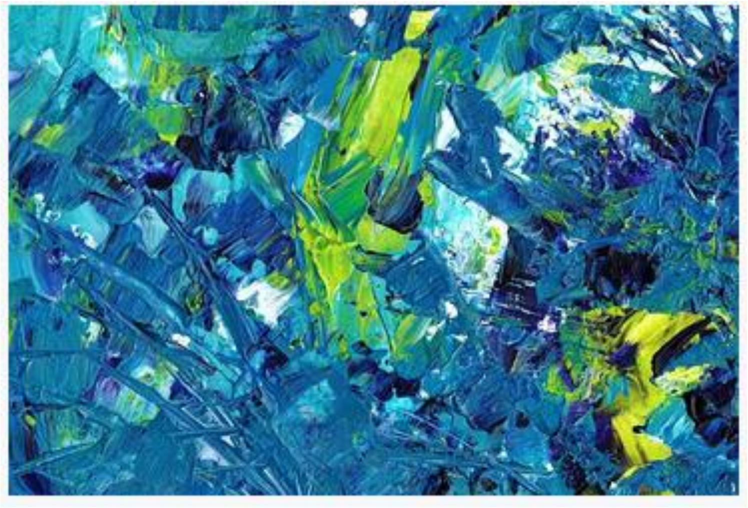
# Averaging over a Forest

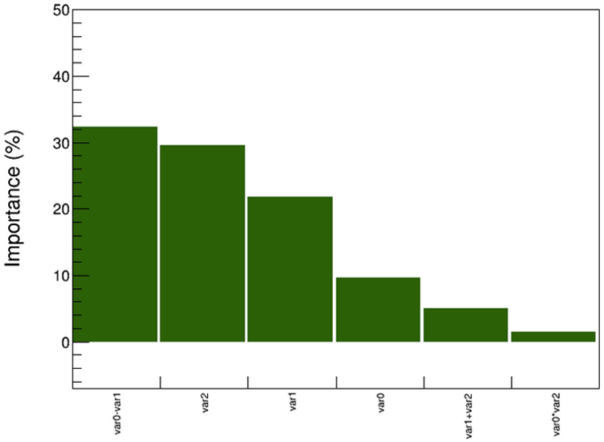


# H to ZZ to 4Leptons



# Feature Selection





# Classical Feature Selection

**In data analysis one of the most crucial decisions is which features to use**

- Garbage In = Garbage Out

**Main Ingredients:**

- **Relevance** to the problem
- How well feature is **understood**
- Its **power** and **relationship** with others

# Typical Initial Set

**Basic measurements covering phase space of problem:**

- Functions made from them

**More complex features using domain knowledge to help discriminate among classes**

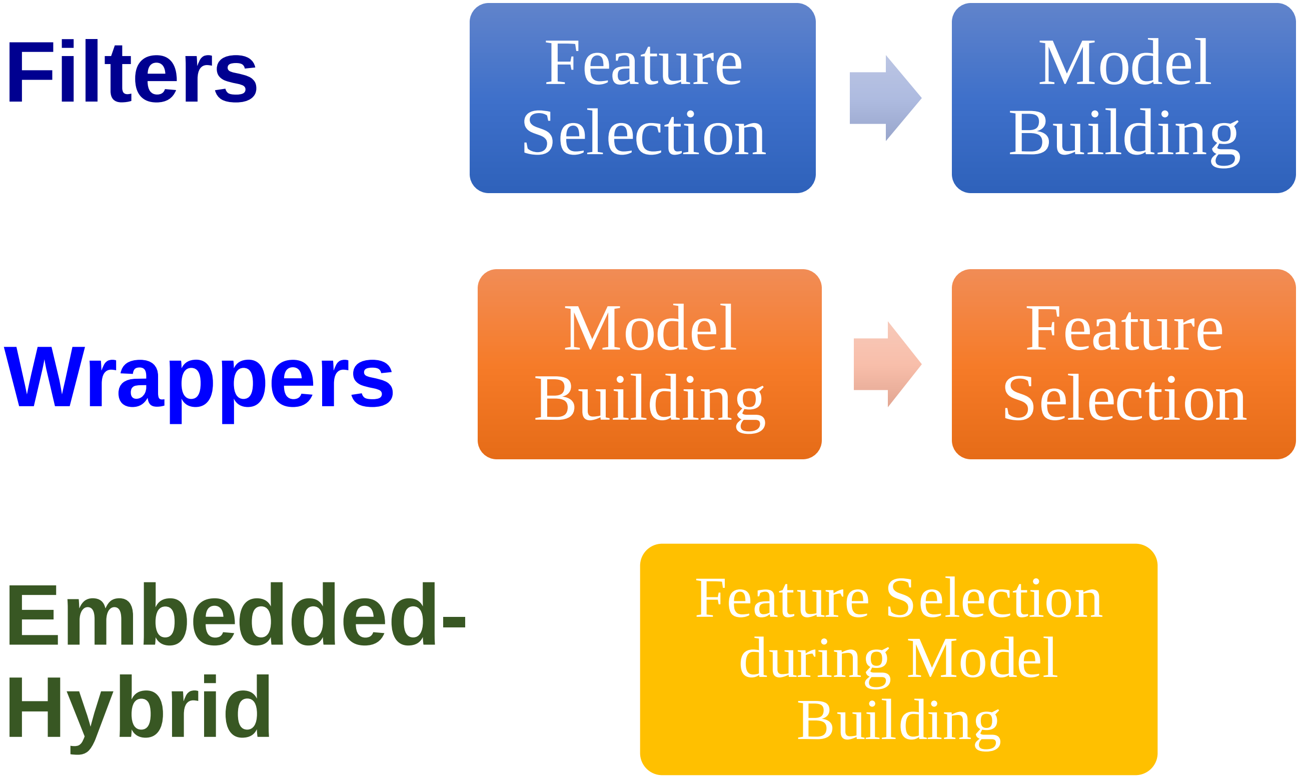
- 1-D discriminants

# Feature Engineering

## Combining features with each other

- this set can grow quickly
- balance between
  - ***Occam's razor***
  - **Need for additional performance**

# Feature Selection Methods



**Filters**

Feature Selection → Model Building

**Wrappers**

Model Building → Feature Selection

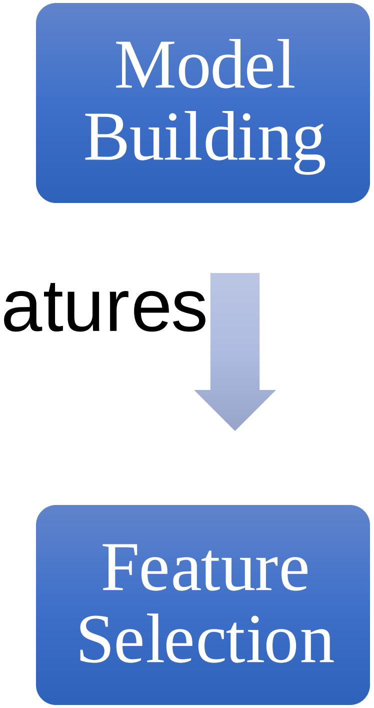
**Embedded-Hybrid**

Feature Selection during Model Building

# Wrapper Methods

## Selection tied to a model:

- More accurate
- Assess feature interactions
- Search for optimal subset of features



## Types:

- **Methodical**
- **Probabilistic**
  - random hill-climbing
- **Heuristic**
  - forward backward elimination

# Example Wrapper

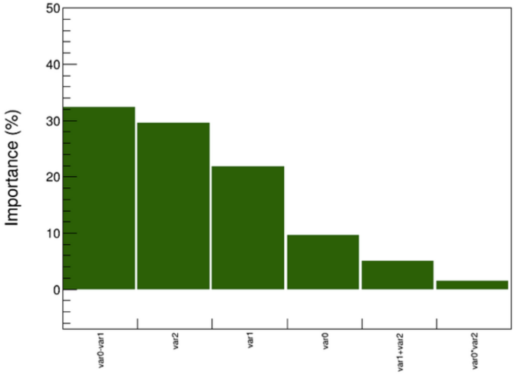
**Feature Importance** ⟶ proportional to **classifier performance** in which feature participates

$$FI(X_i) = \sum_{S \subseteq V : X_i \in S} F(S) \times W_{X_i}(S)$$

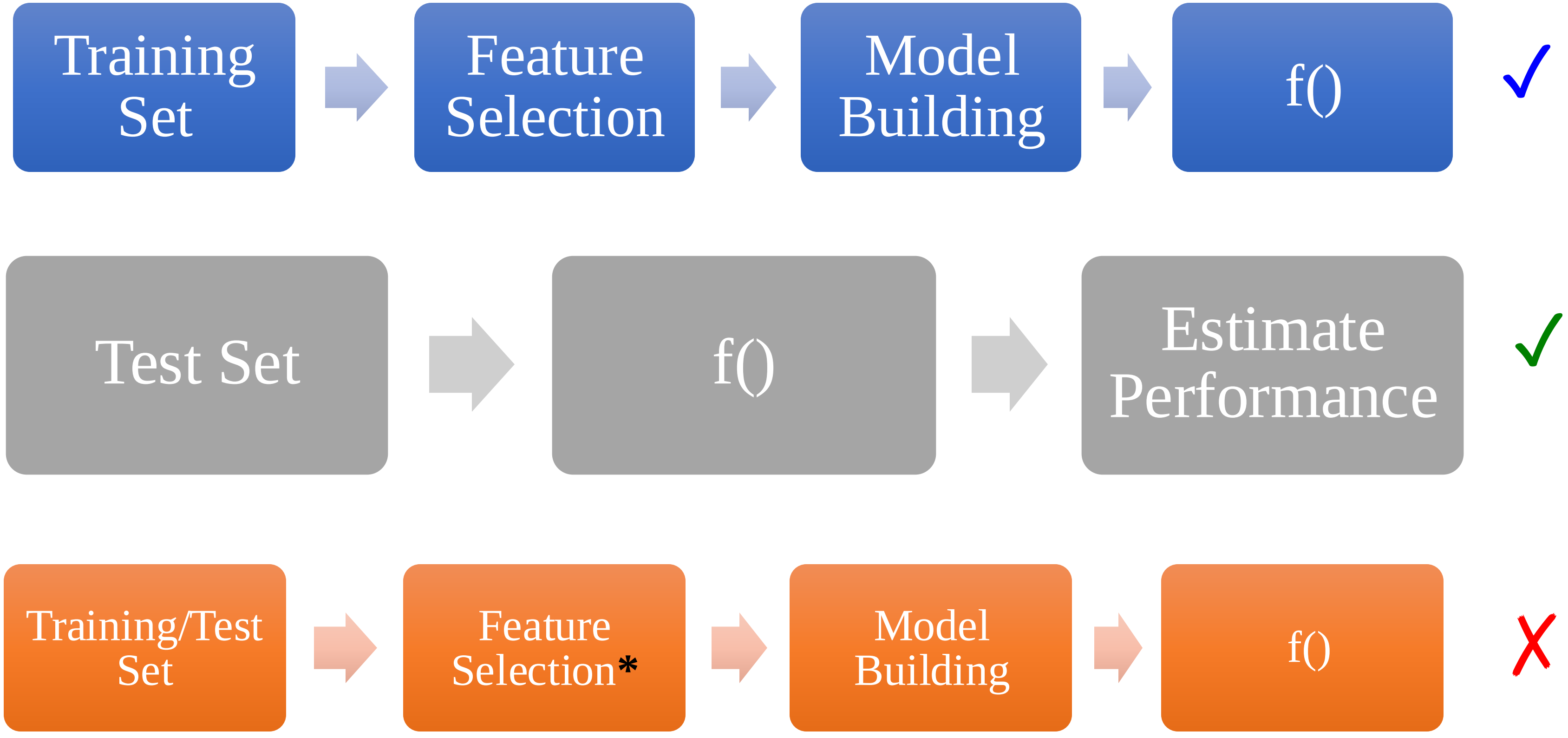
- **Full feature set {V}**
- **Feature subsets {S}**
- **Classifier performance F(S)**

$$W_{X_i}(S) \equiv 1 - \frac{F(S - \{X_i\})}{F(S)}$$

- Stochastic version
  uses random subset seeds



# Practicum



Training Set → Feature Selection → Model Building → f() ✓

Test Set → f() → Estimate Performance ✓

Training/Test Set → Feature Selection* → Model Building → f() ✗

***Feature Selection Bias**

# Embedded Methods

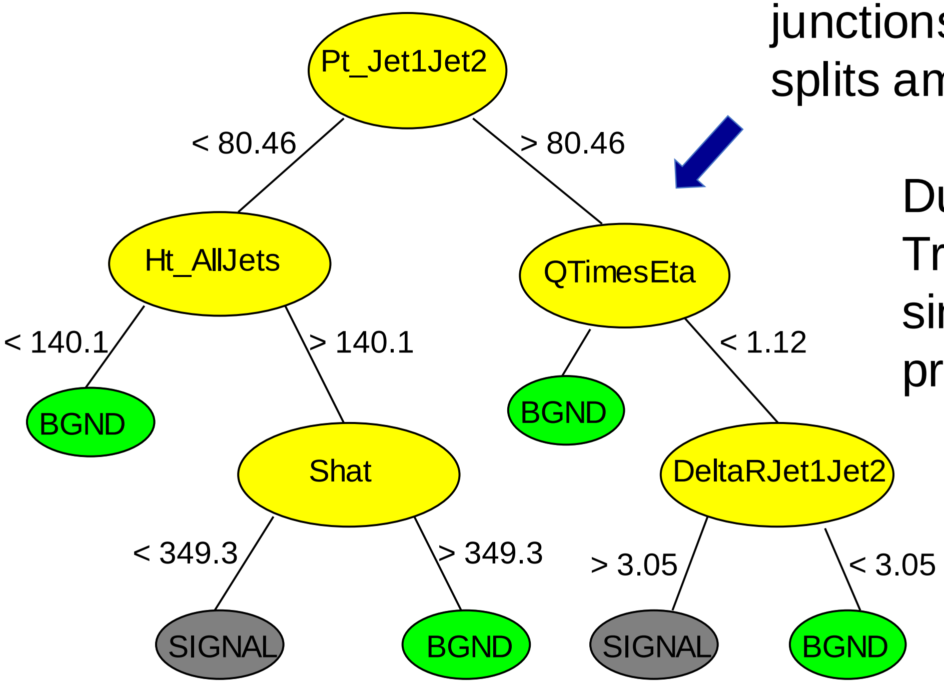
## Incorporate feature importance in the model-building process

- **Penalize features** in the classification or regression process
- **Regularization**
  - LASSO
  - Regularized Trees

# Regularized Trees

Inspired by J. Friedman and Popescu, 2008 work on rules regularization

## Decision Tree:



**Votes** taken at decision junctions on possible splits among the features

During voting Regularized Trees **penalize** features similar to those used in previous decisions

End up with a **high quality** feature set