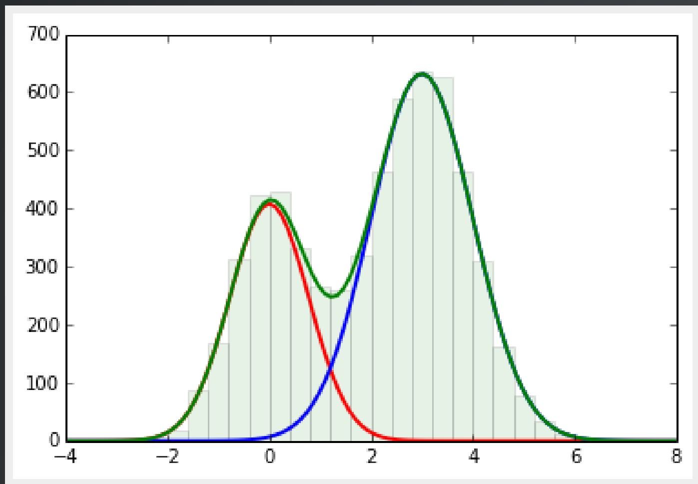
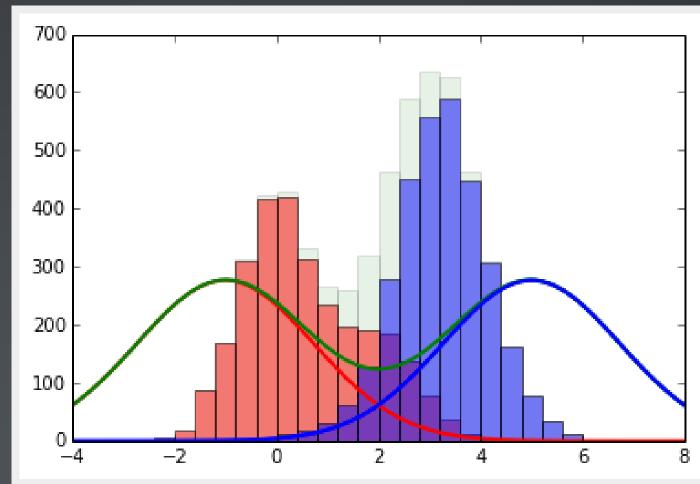
MACHINE LEARNING IN HIGH ENERGY PHYSICS

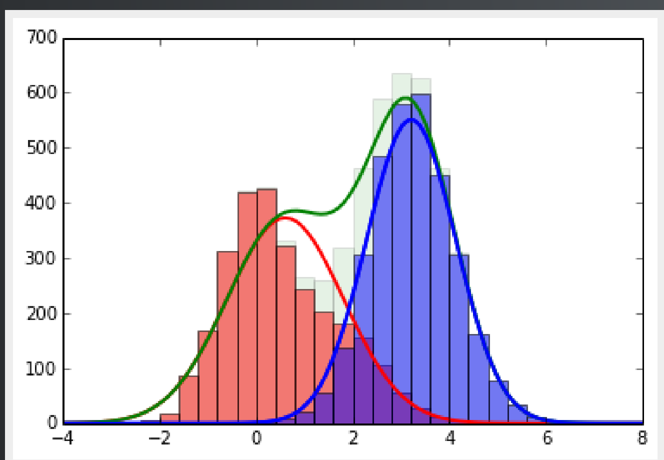
PRACTICAL CLASS #2

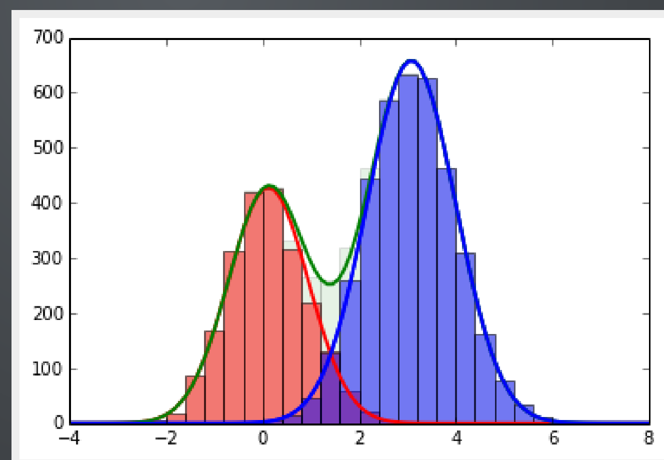
Alex Rogozhnikov, 2015

EXPECTATION-MAXIMIZATION (EM)





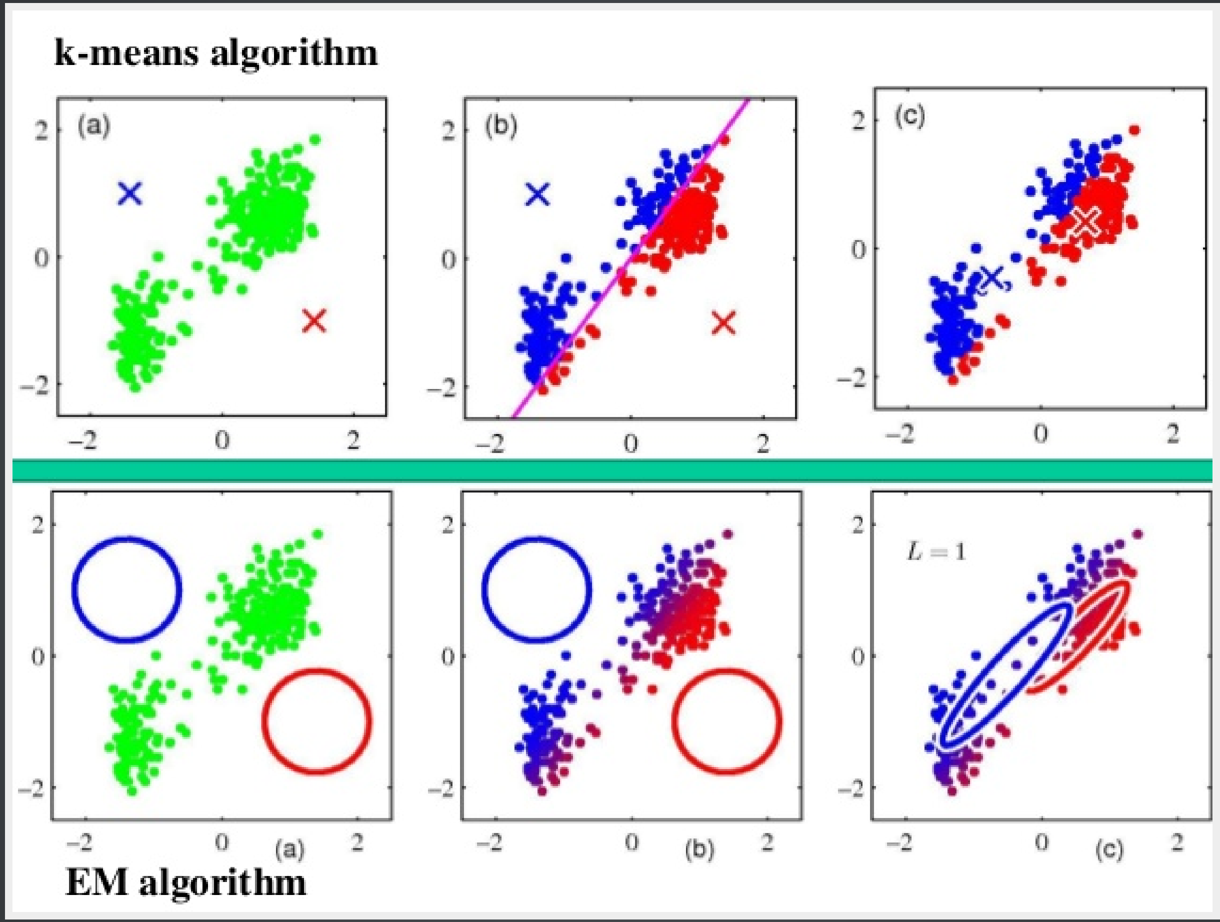




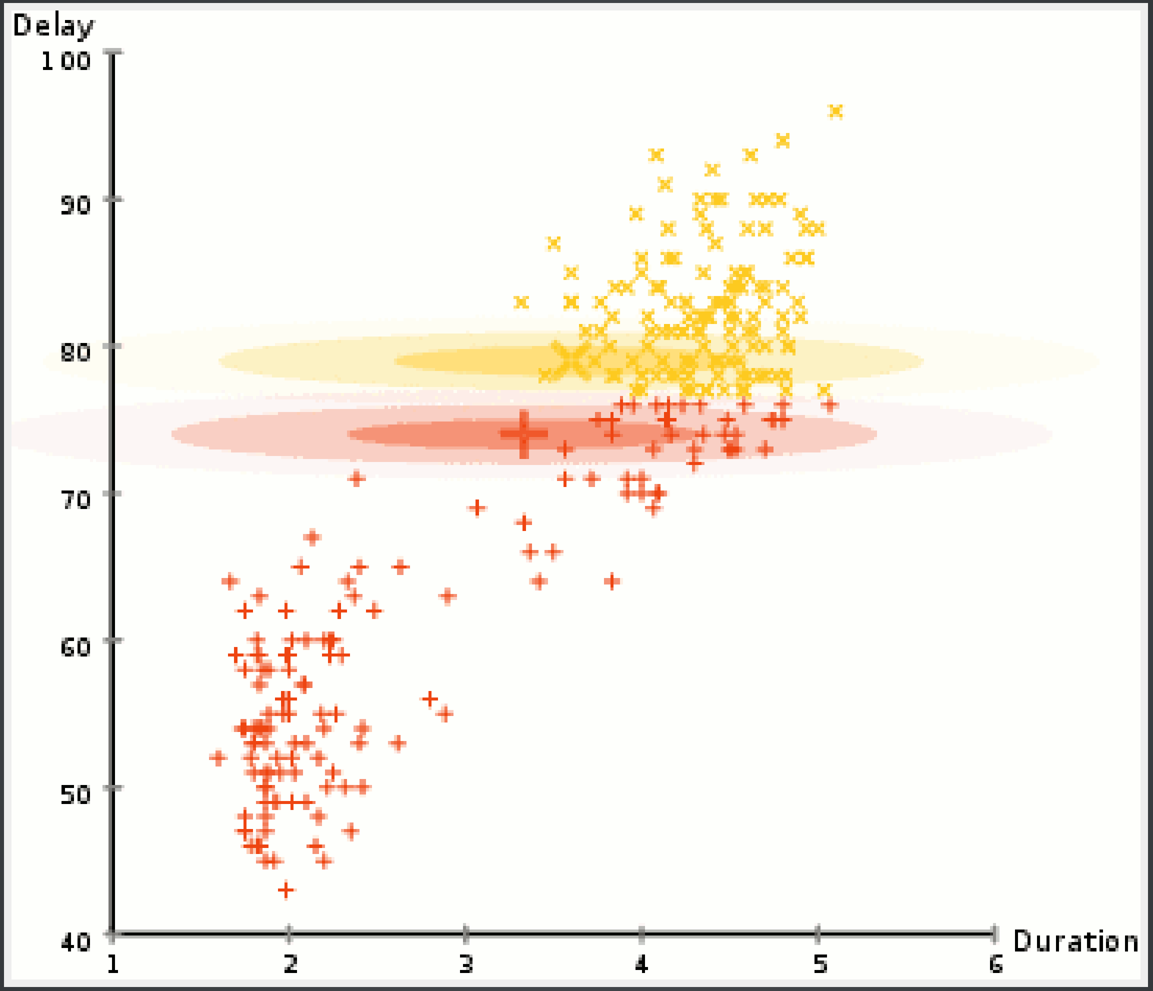
after 1 step

after 15 step

K-MEANS AND EM

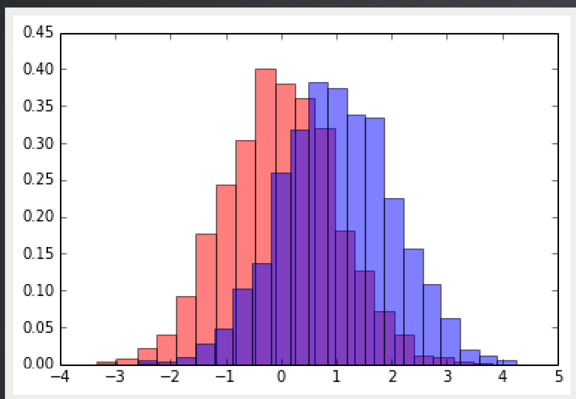


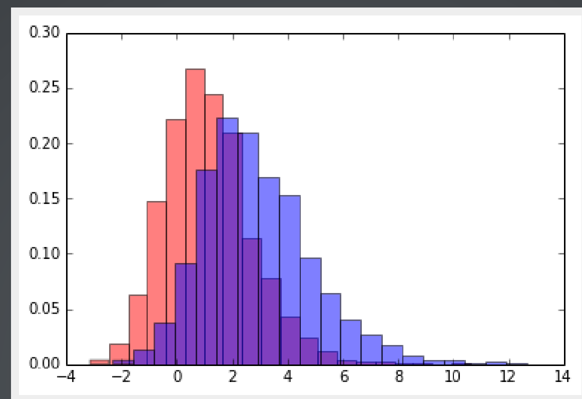
EXPECTATION-MAXIMIZATION (EM)

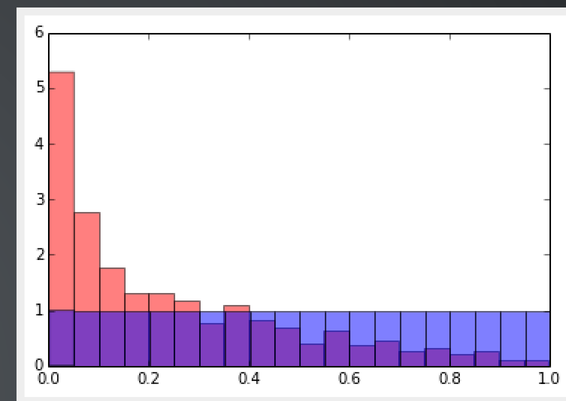


ROC CURVE AND MEASUREMENT OF QUALITY

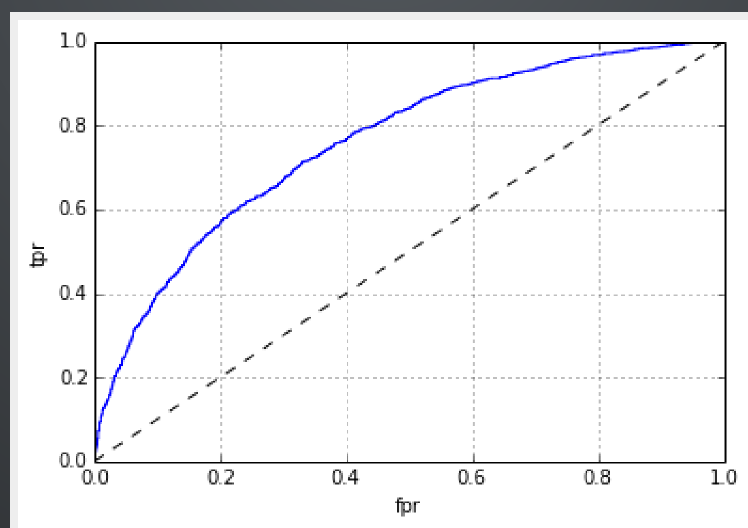
The classifier's output in binary classification is real variable







These distributions have same roc-curve:



ROC CURVE

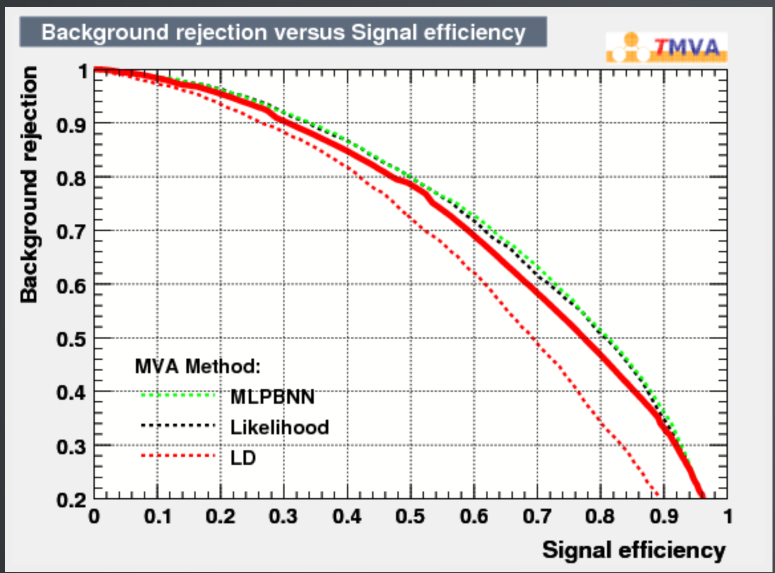
- Contains important information:
  all possible combinations of signal and background efficiencies
  you may achieve by setting threshold
- Particular values of thresholds (and initial pdfs) don't matter,
  ROC curve doesn't contain this information
- ROC curve = information about order of events:

```
s s b s b ...  b b s b b
```

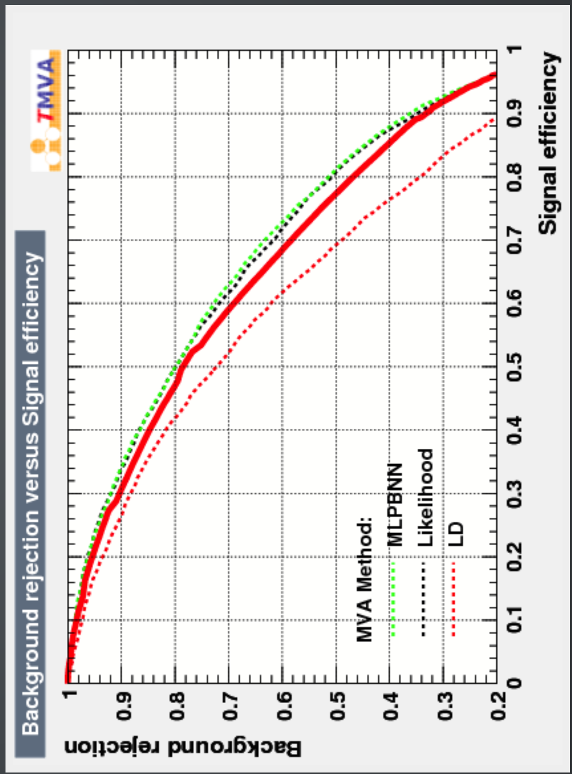
- Comparison of algorithms should be based on information
  from ROC curve

TERMINOLOGY AND CONVENTIONS

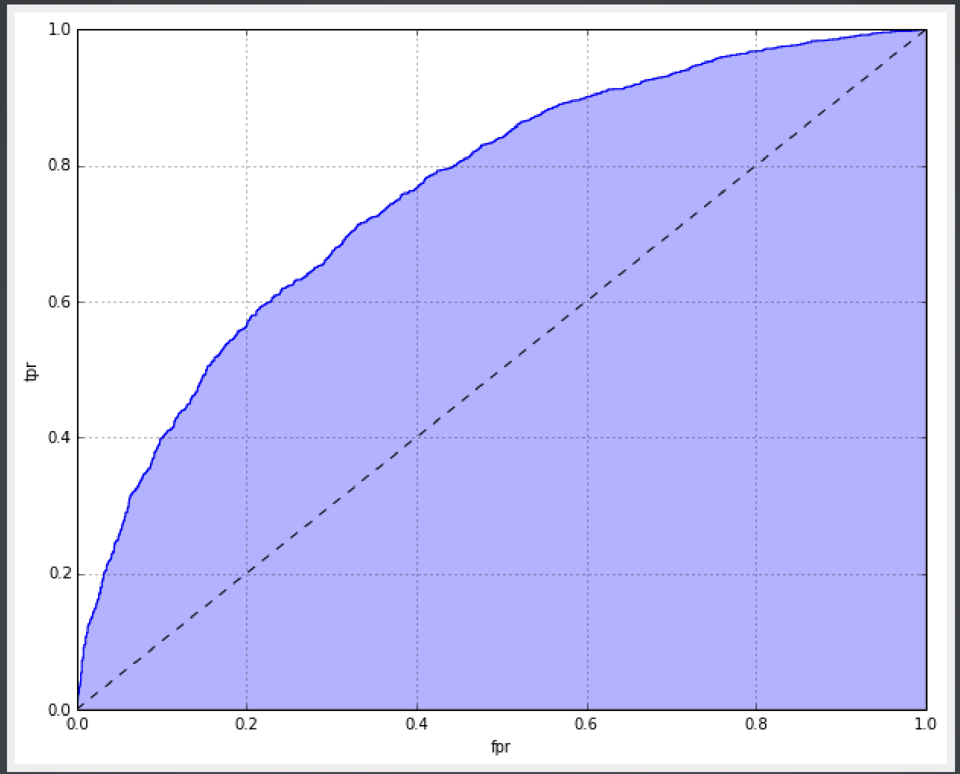
- fpr = background efficiency = b
- tpr = signal efficiency = s



→



ROC AUC (AREA UNDER THE ROC CURVE)

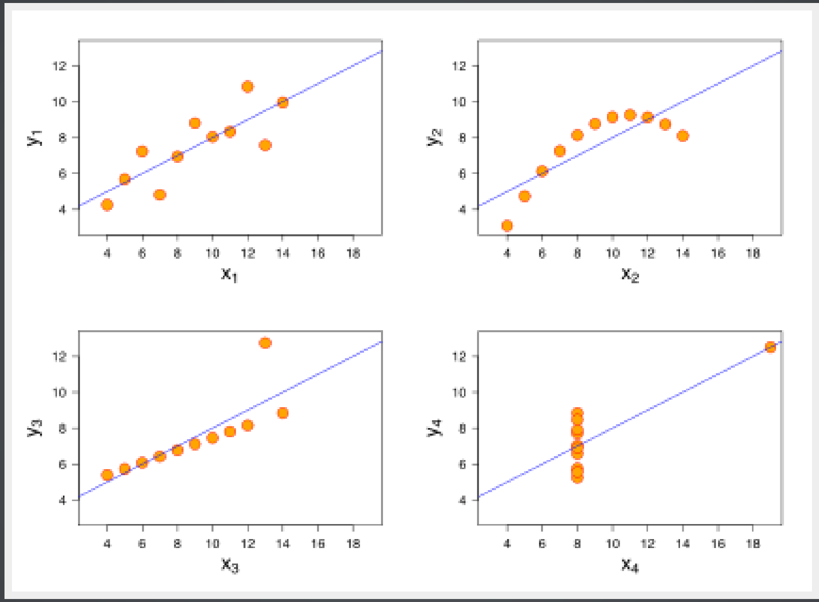


$$ROC\ AUC = P(x < y)$$

where x, y are predictions of random background and signal events.

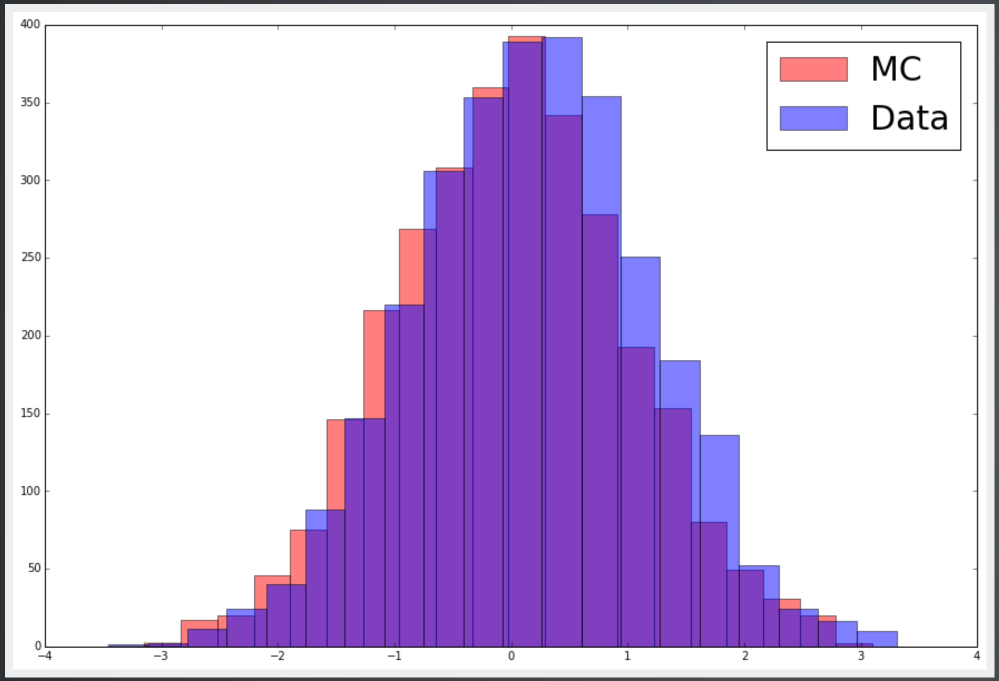
COMPARISON OF MULTIDIMENSIONAL DISTRIBUTIONS

- Usually 1d/2d distributions are compared
- But multidimensional data has much more complex



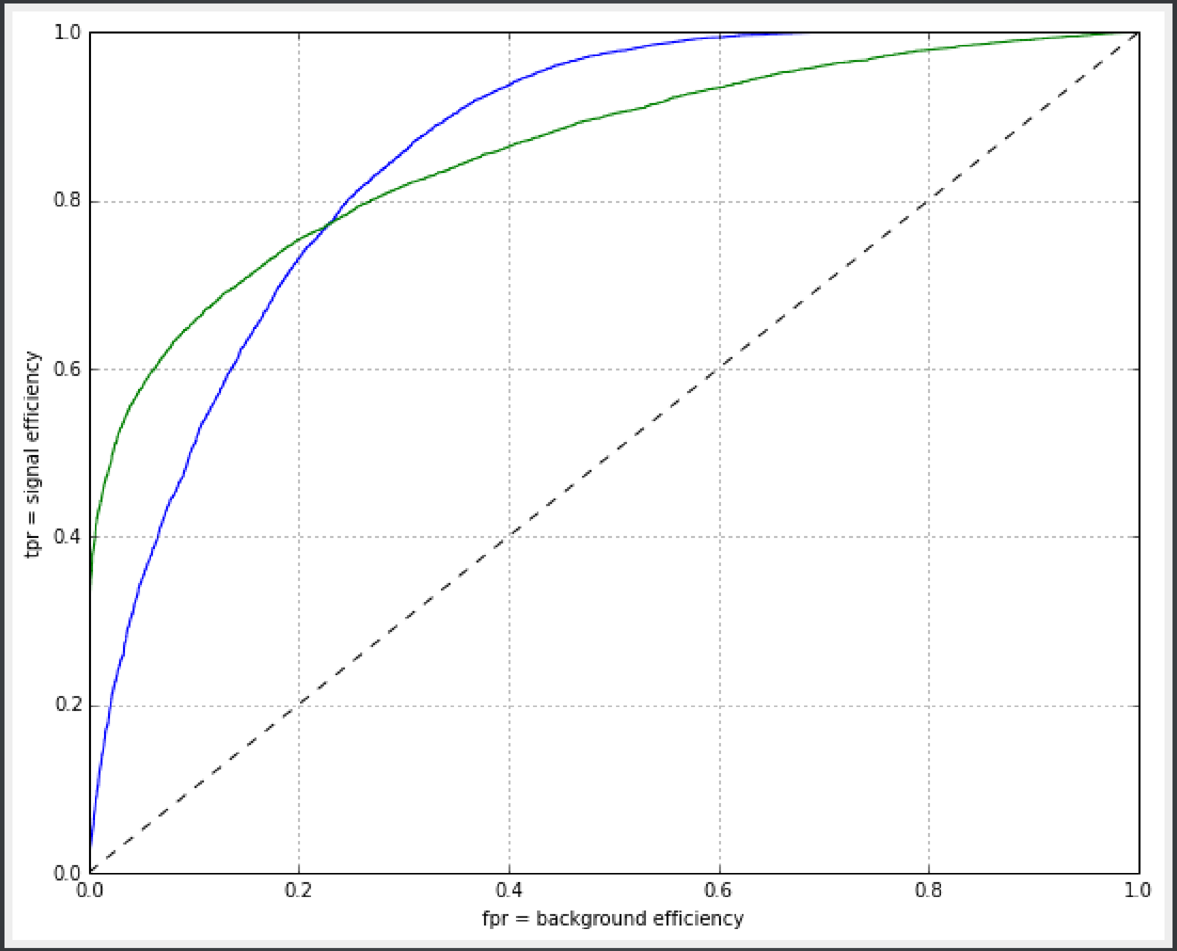
COMPARISON OF MULTIDIENSIONAL DISTRIBUTIONS

- ROC AUC is good as statistics to test whether classifier can distinguish your datasets



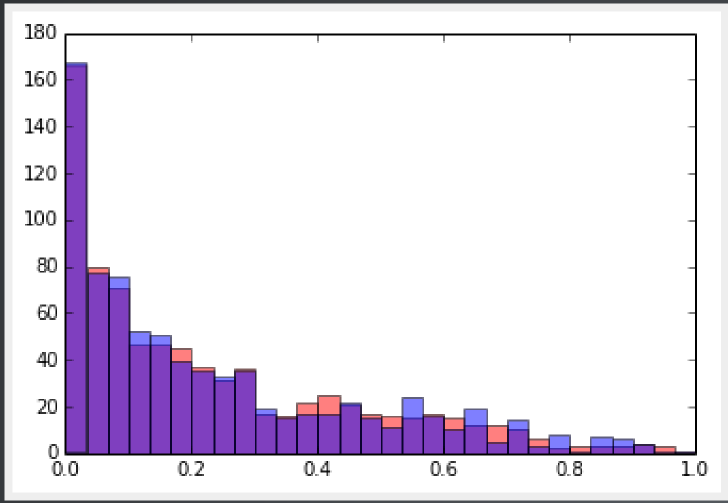
- Significance? Use U-test!

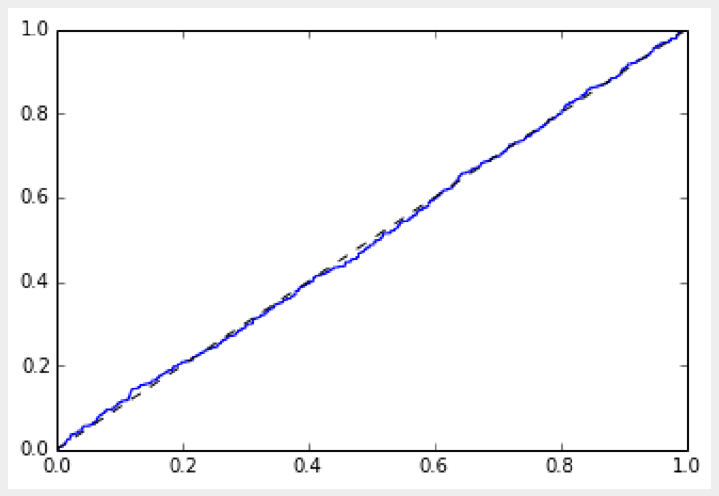
Which classifier is better for triggers?
(they have same ROC AUC)



Usage of specific metrics: Punzy, AMS, tpr at fixed fpr

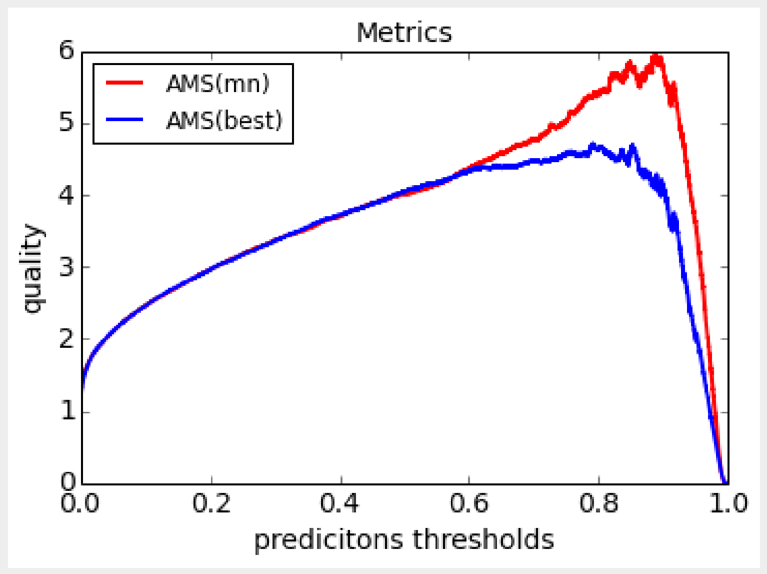
P-P PLOT, COMPARISON OF DISTRIBUTIONS





AMS VS CUT

$$\text{AMS}^2 = 2\left((s+b)\log(1+\frac{s}{b}) - s\right)$$



Check that your metrics is stable (on subsets of test dataset) (cut-based metrics require much more events).

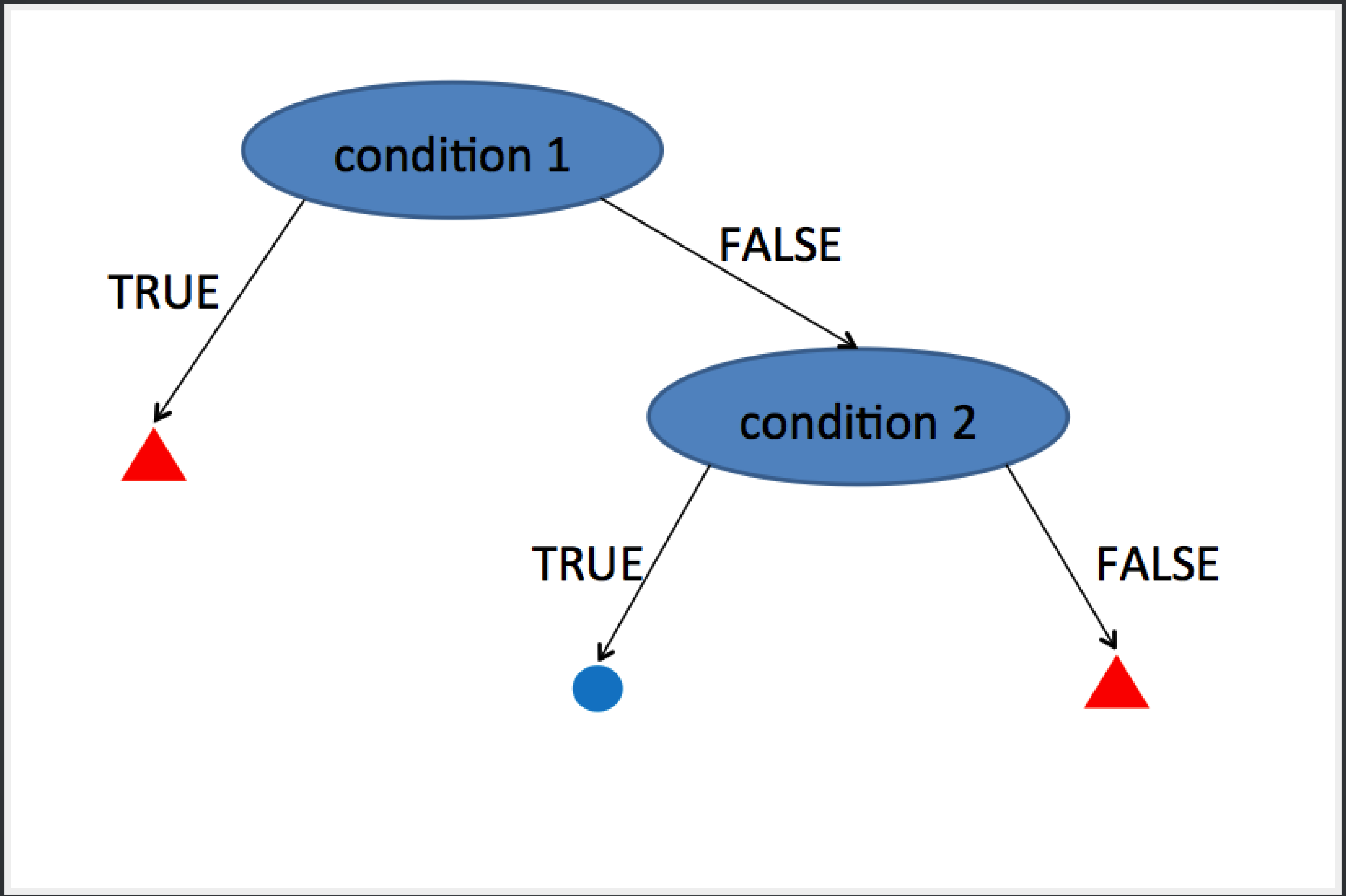
WP-based metrics are unstable and require much more events

https://indico.cern.ch/event/316800/material/slides/0.pdf

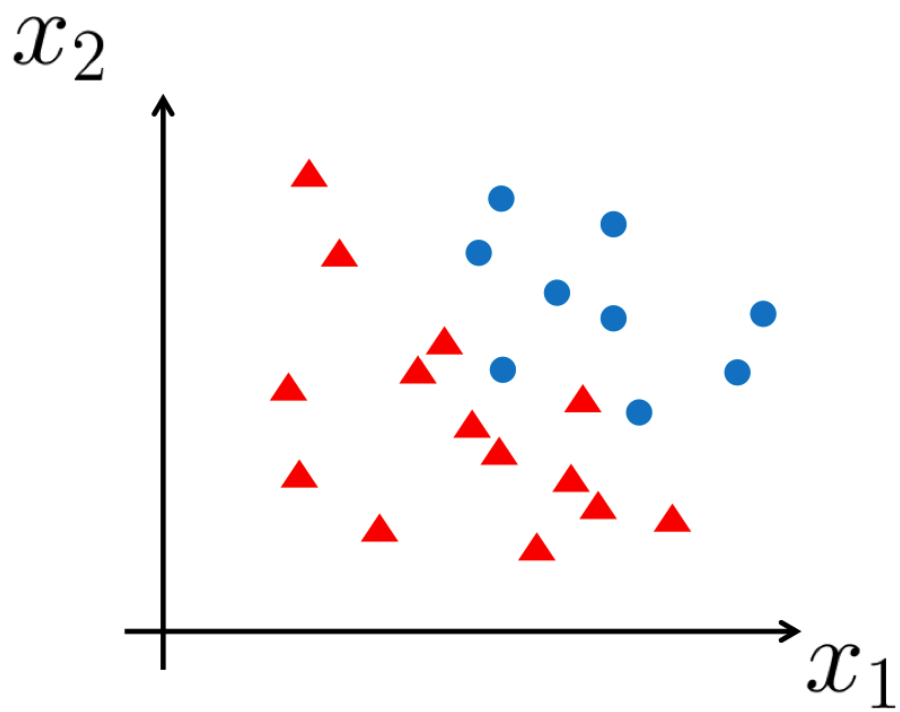
Kaggle discussion on regularized AMS

KNN DEMONSTRATION

DECISION TREES: IDEA

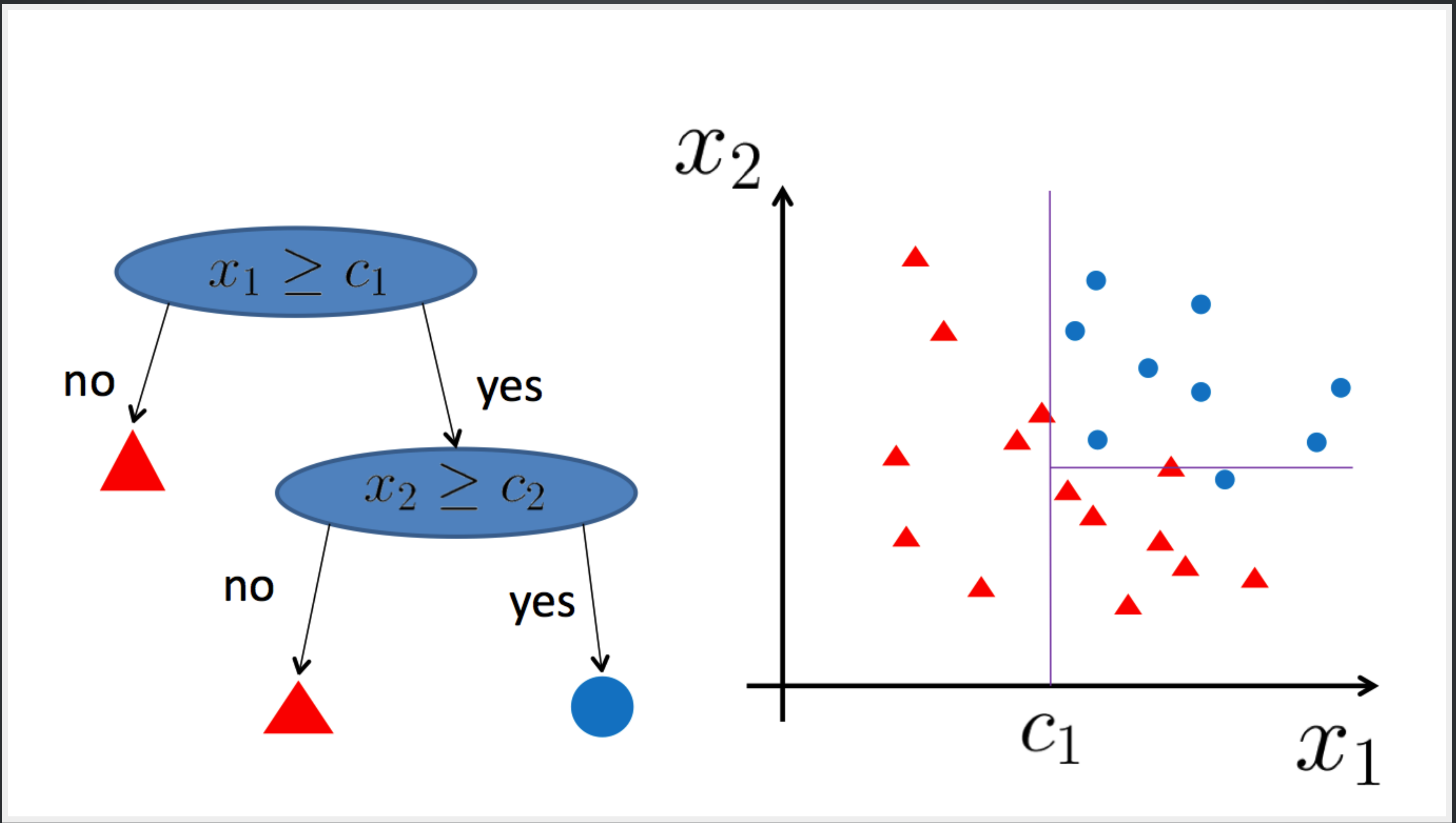


DECISION TREES



"Stump" conditions: x > c

DECISION TREES



PURITY OF NODE

How to select best split?

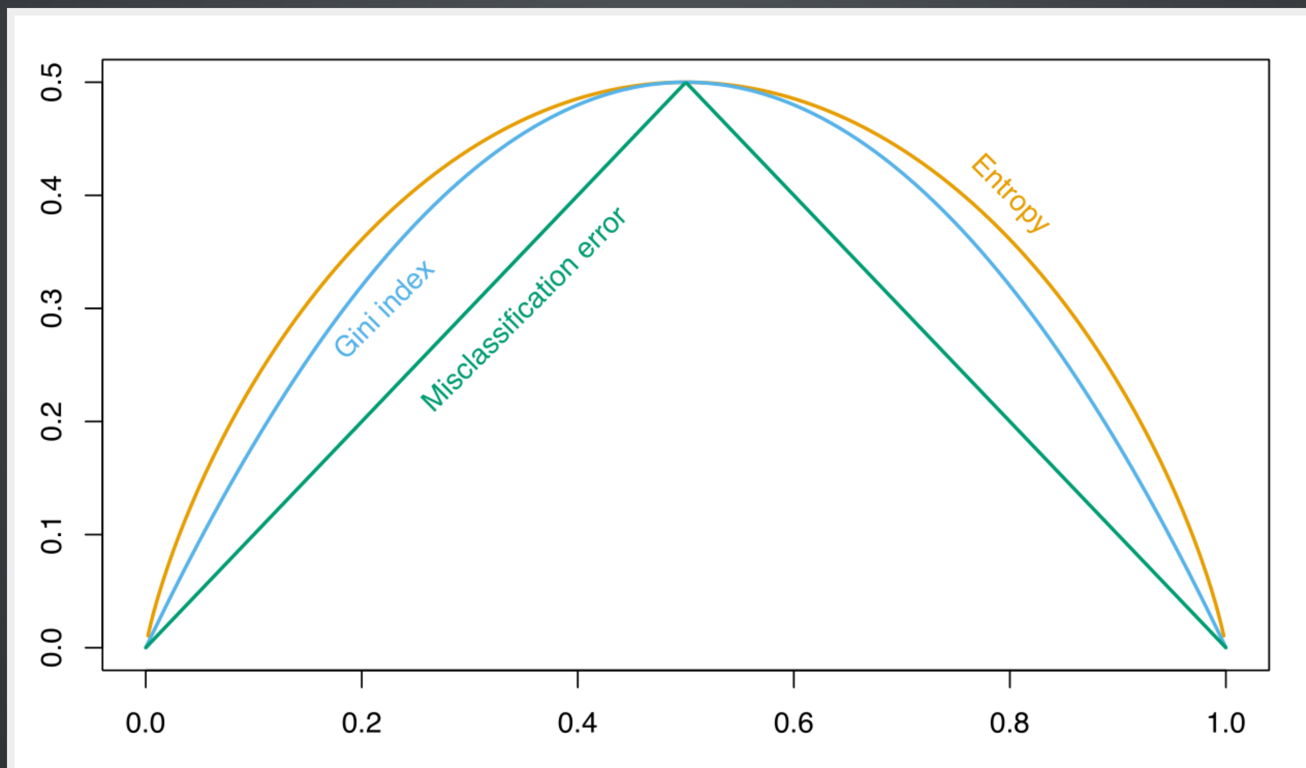
Let p_0, p_1 be portions of signal and background in leaf

- Error: $Q = \frac{1}{N} \sum_{i=1}^{N} I[y_i \neq \tilde{y}_i]$
- Gini Index: $Q = p_0 p_1$
- Entropy: $Q = -p_0 \log(p_0) - p_1 \log(p1)$

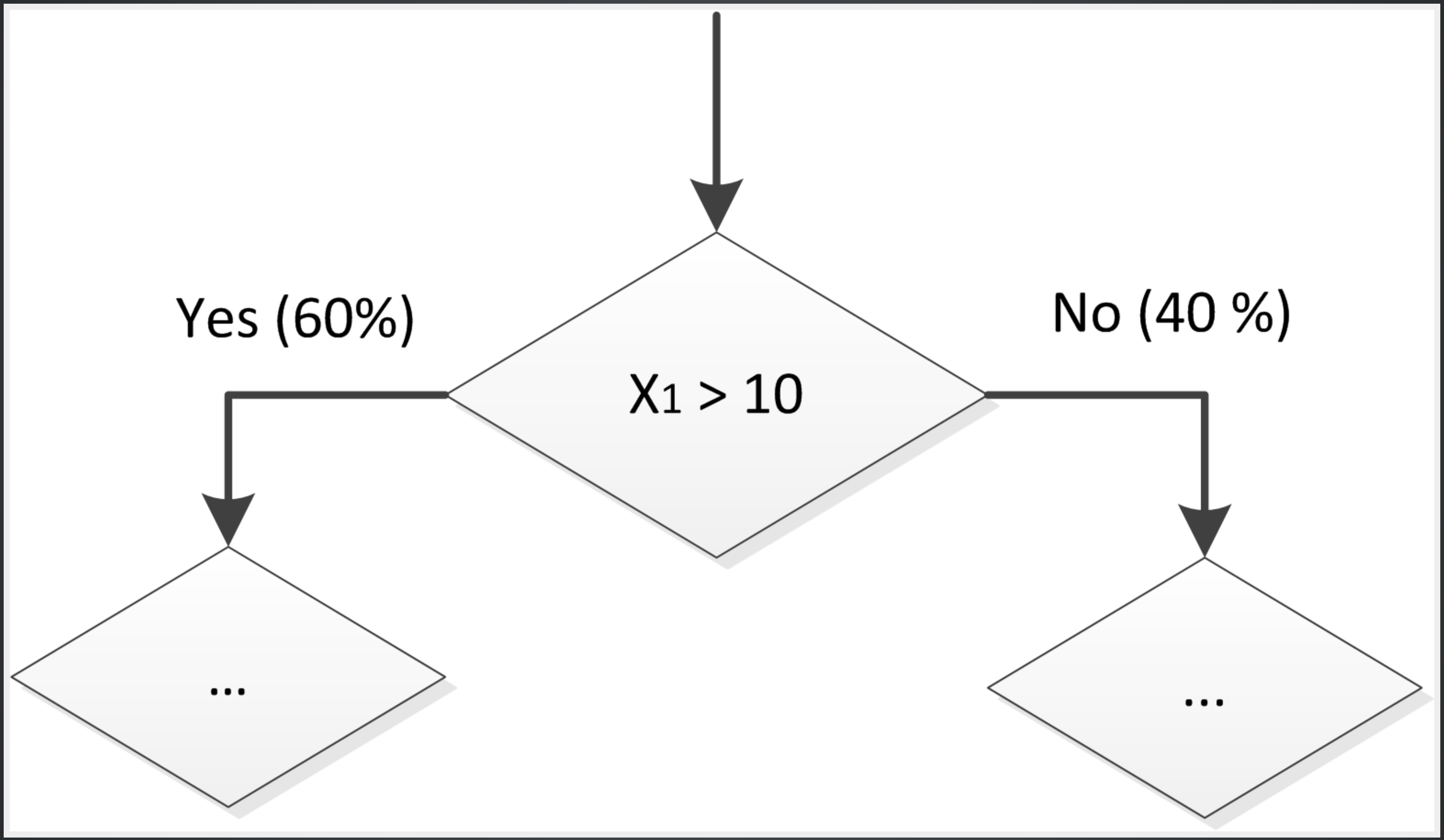
SPLITTING CRITERION

$$\text{Gain} = NQ - N'Q' - N''Q''$$

where N, N', N'' are number of training samples in tree and subtrees, N, N', N'' are node purities respectively

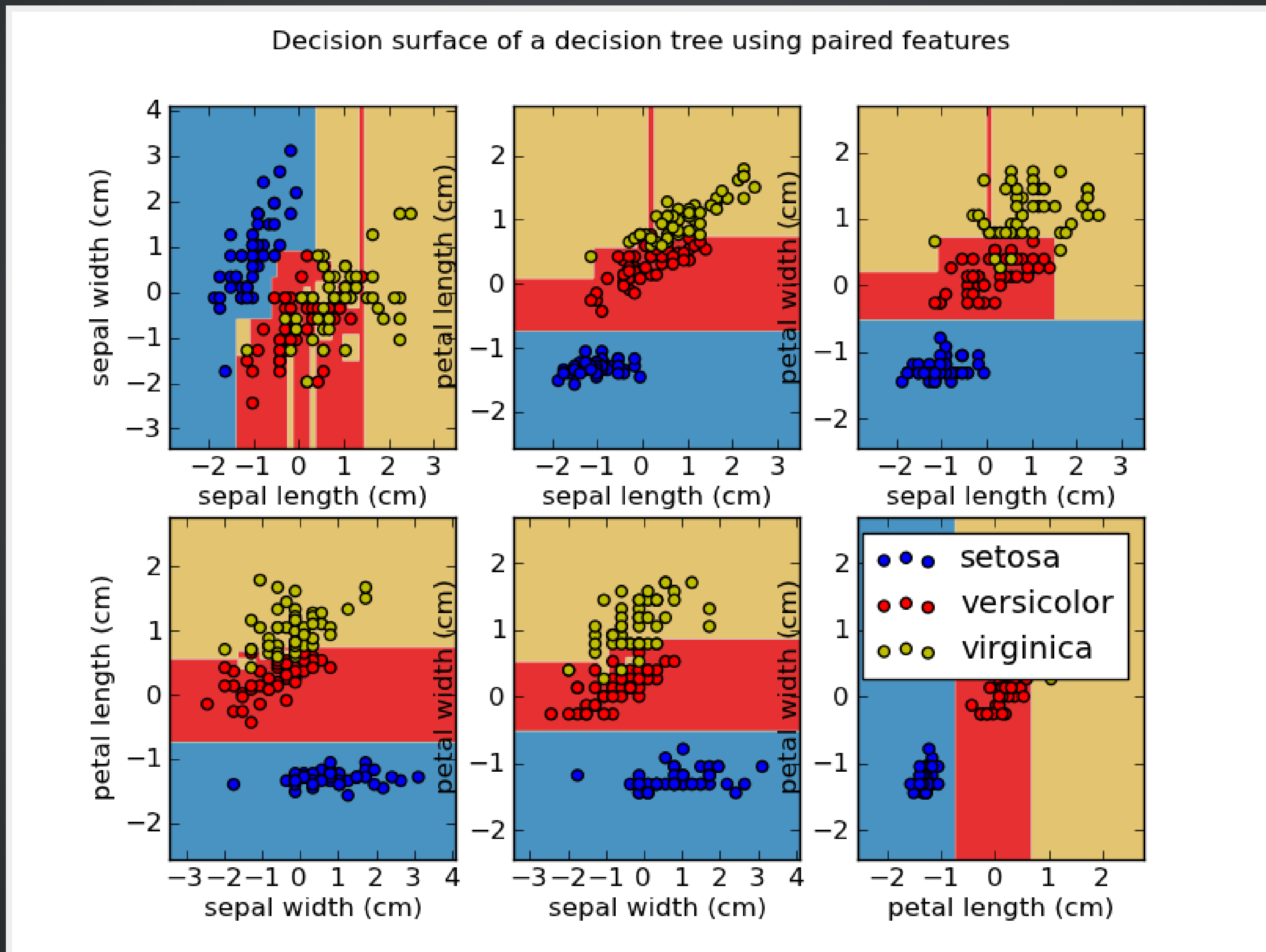


HANDLING OF MISSING VALUES



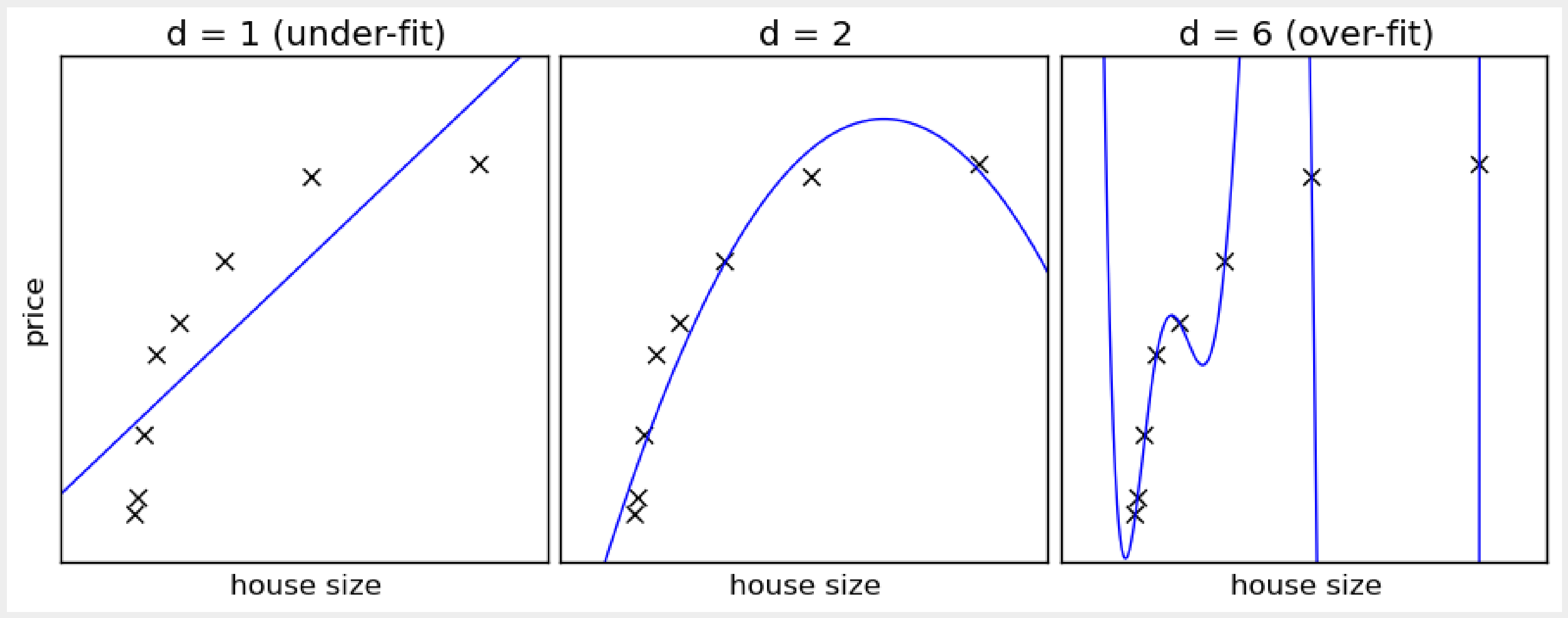
Predict by both subtrees and use prior probabilities

MULTICLASSIFICATION WITH DECISION TREE

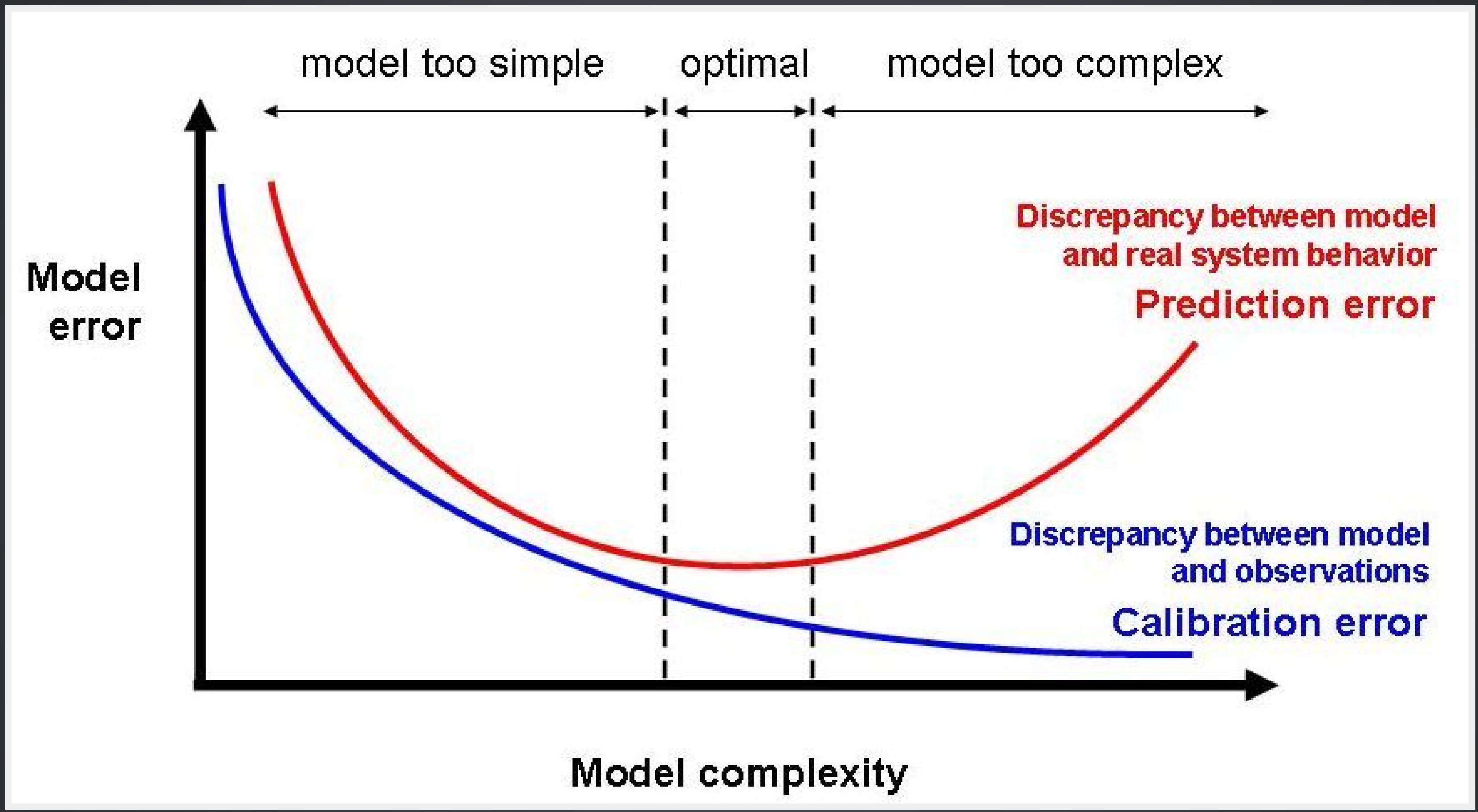


OVERFITTING VS OVERFITTING

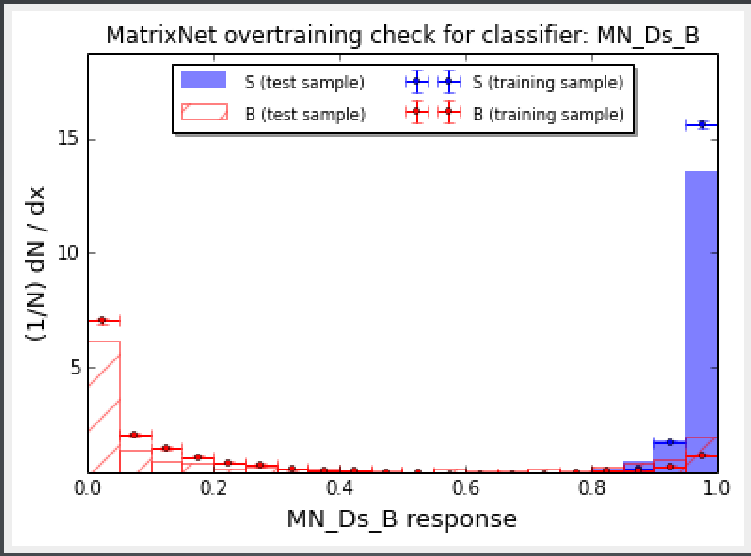
Overfitting in regression (by polynomials)



OVERFITTING VS OVERFITTING

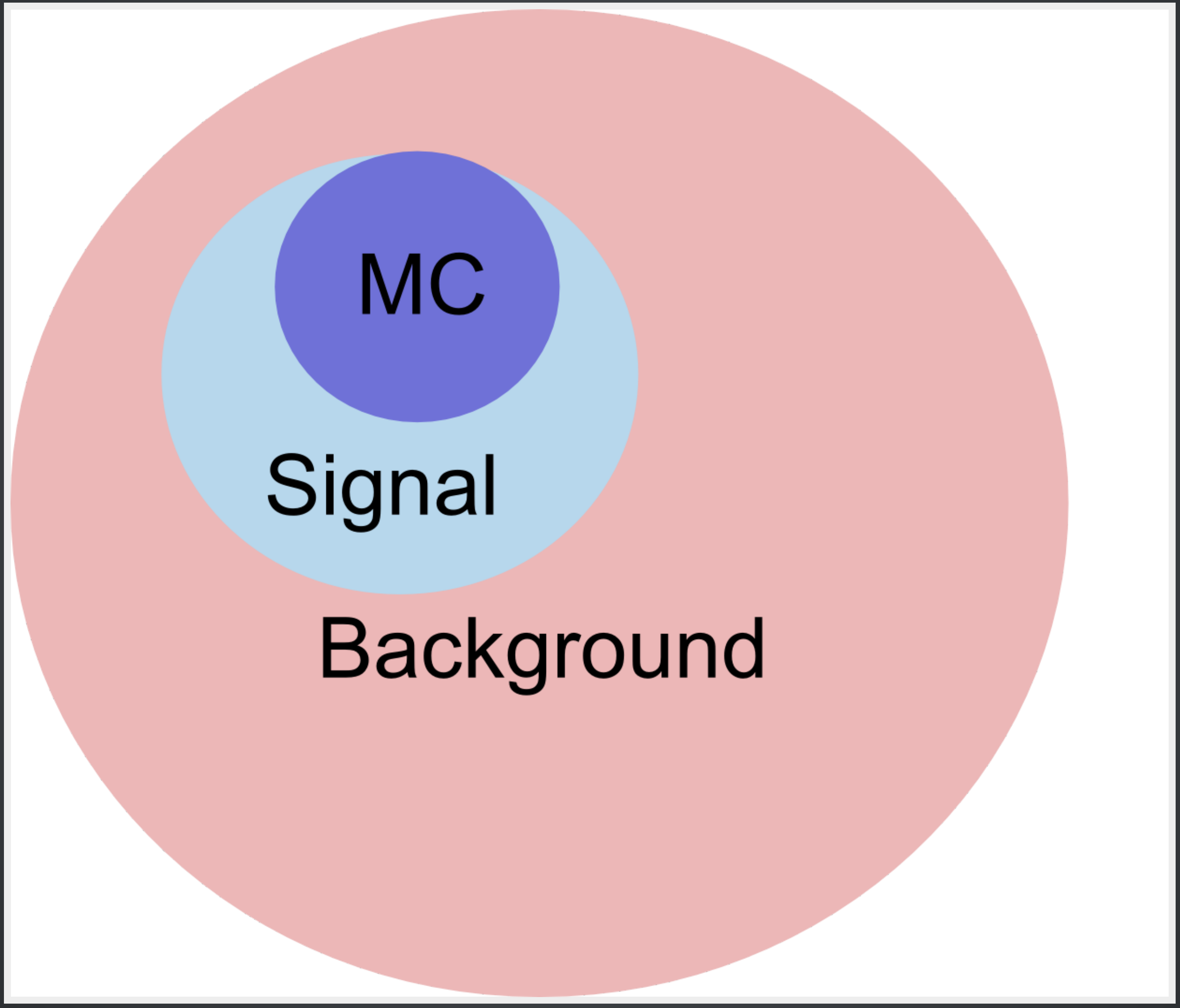


OVERFITTING VS OVERFITTING

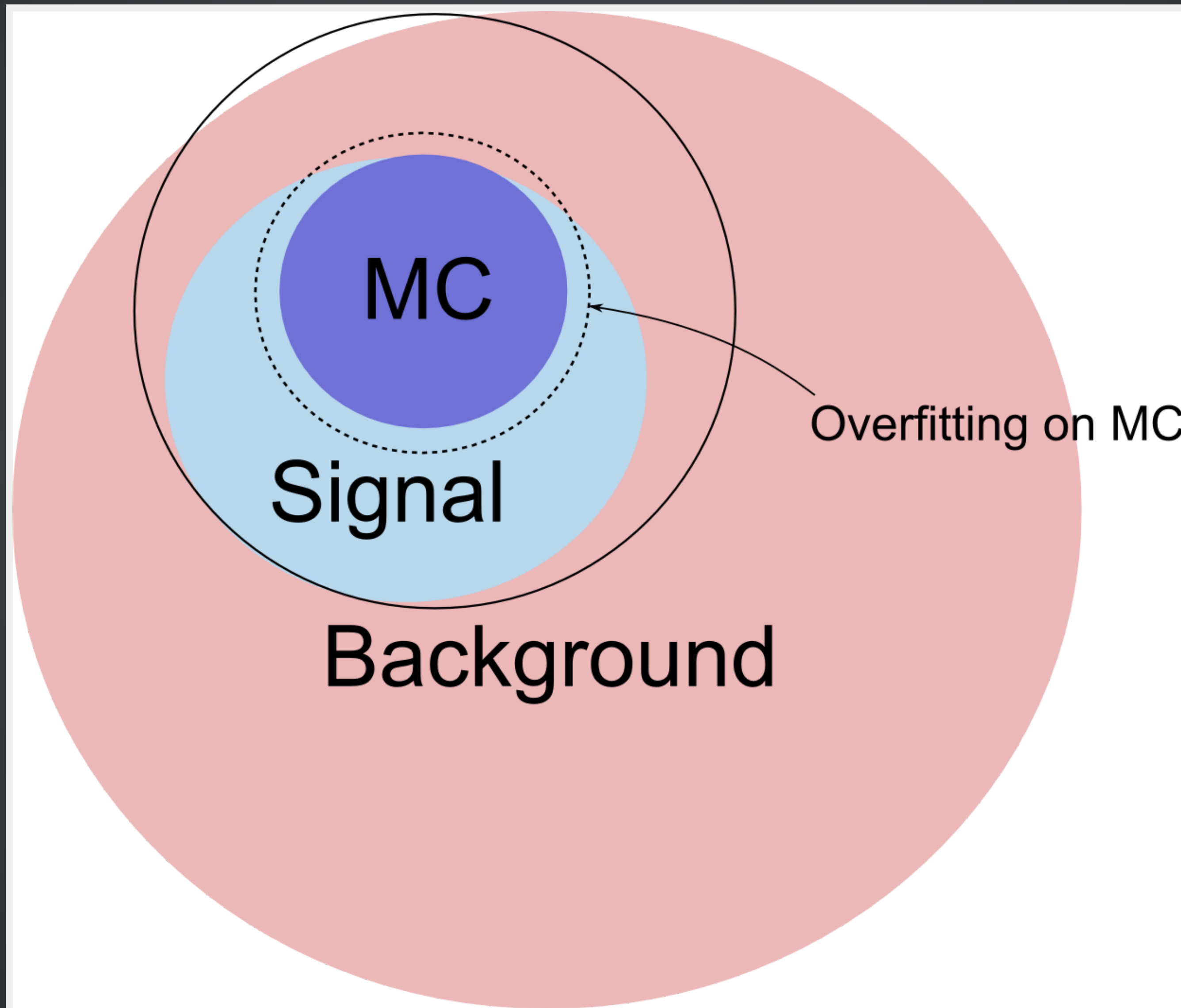


- The situation when model is too complex is called overfitting
- When the quality of prediction on train dataset is much better then on test is called overfitting too.

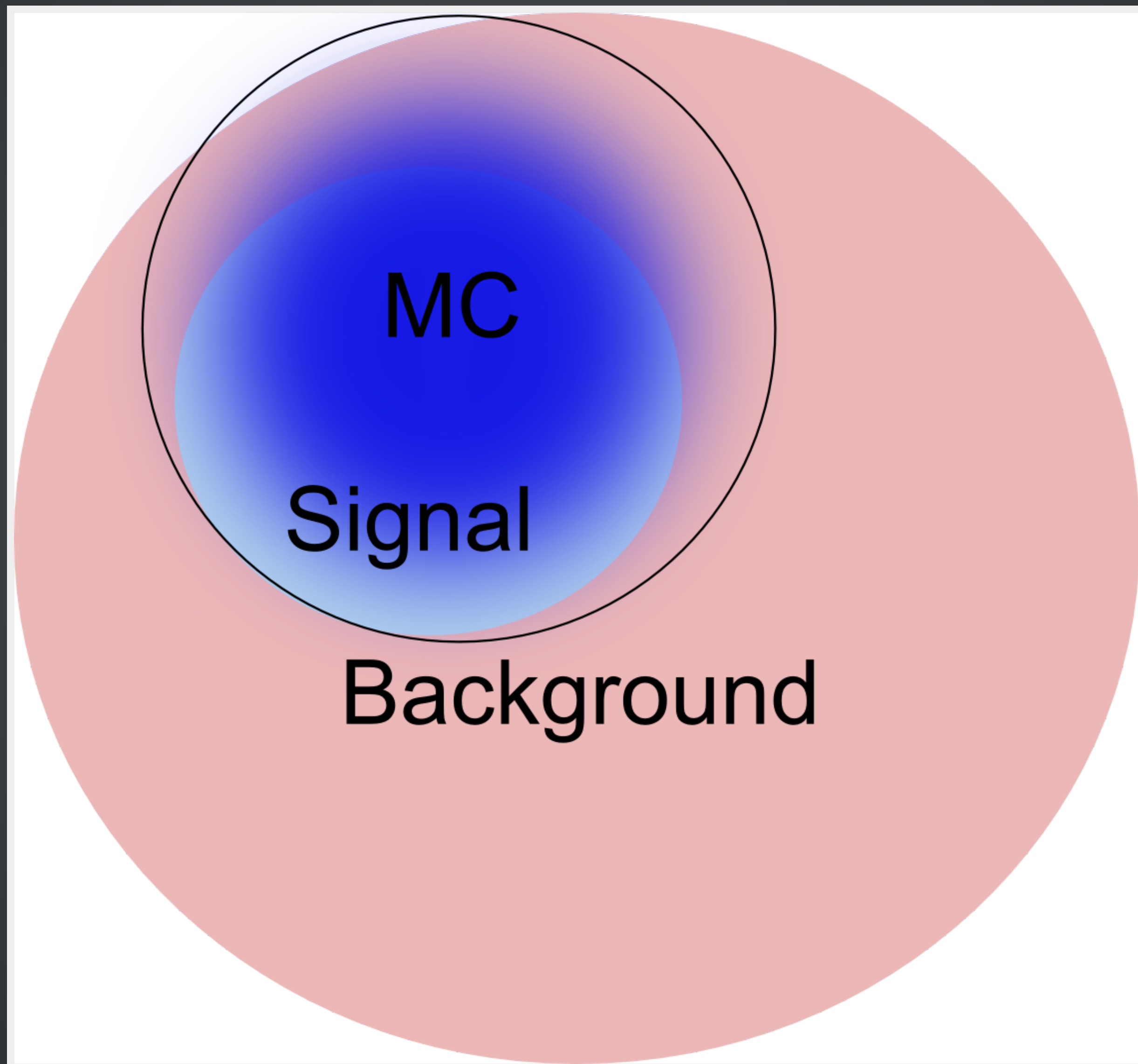
OVERFITTING VS OVERFITTING VS OVERFITTING



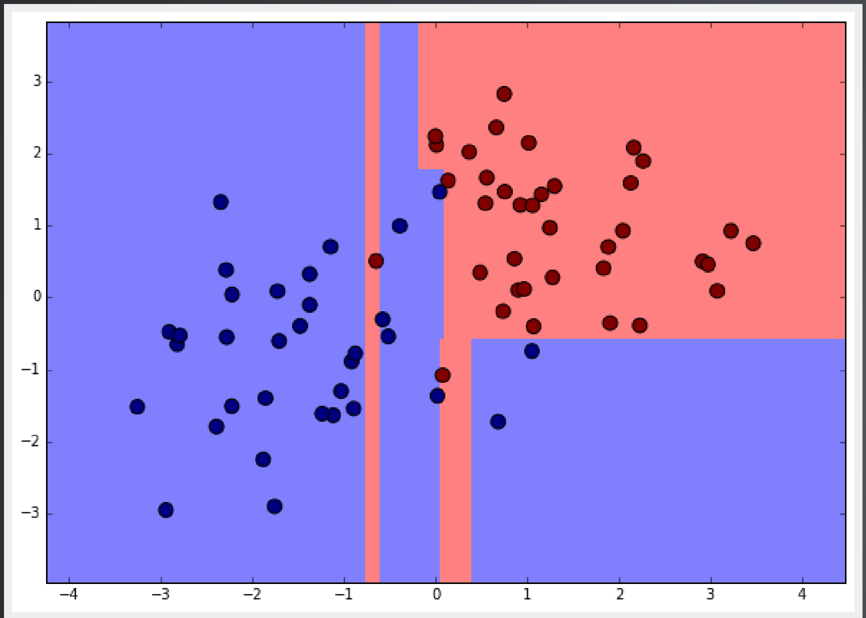
OVERFITTING VS OVERFITTING VS OVERFITTING

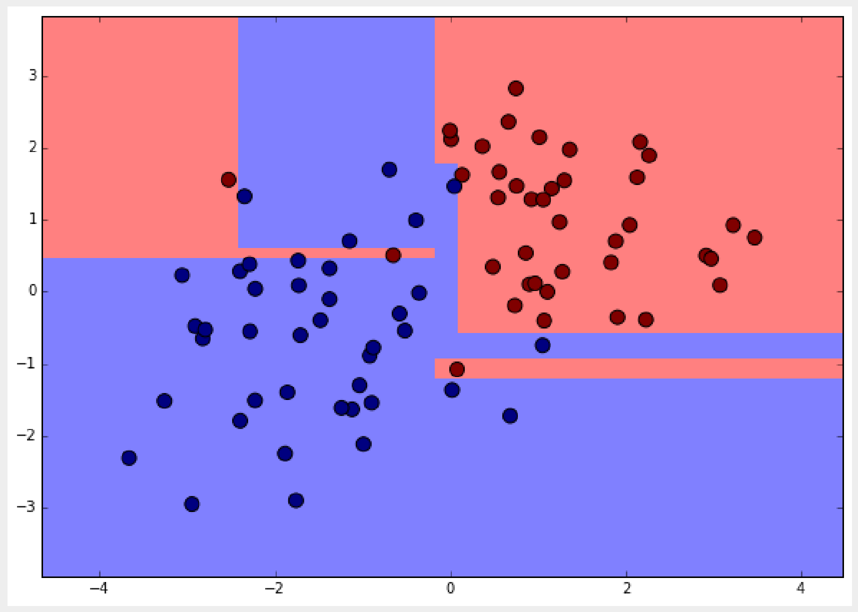


OVERFITTING VS OVERFITTING VS OVERFITTING



DECISION TREE IS UNSTABLE TO SMALL VARIATIONS IN TRAINING SET





DECISION TREE

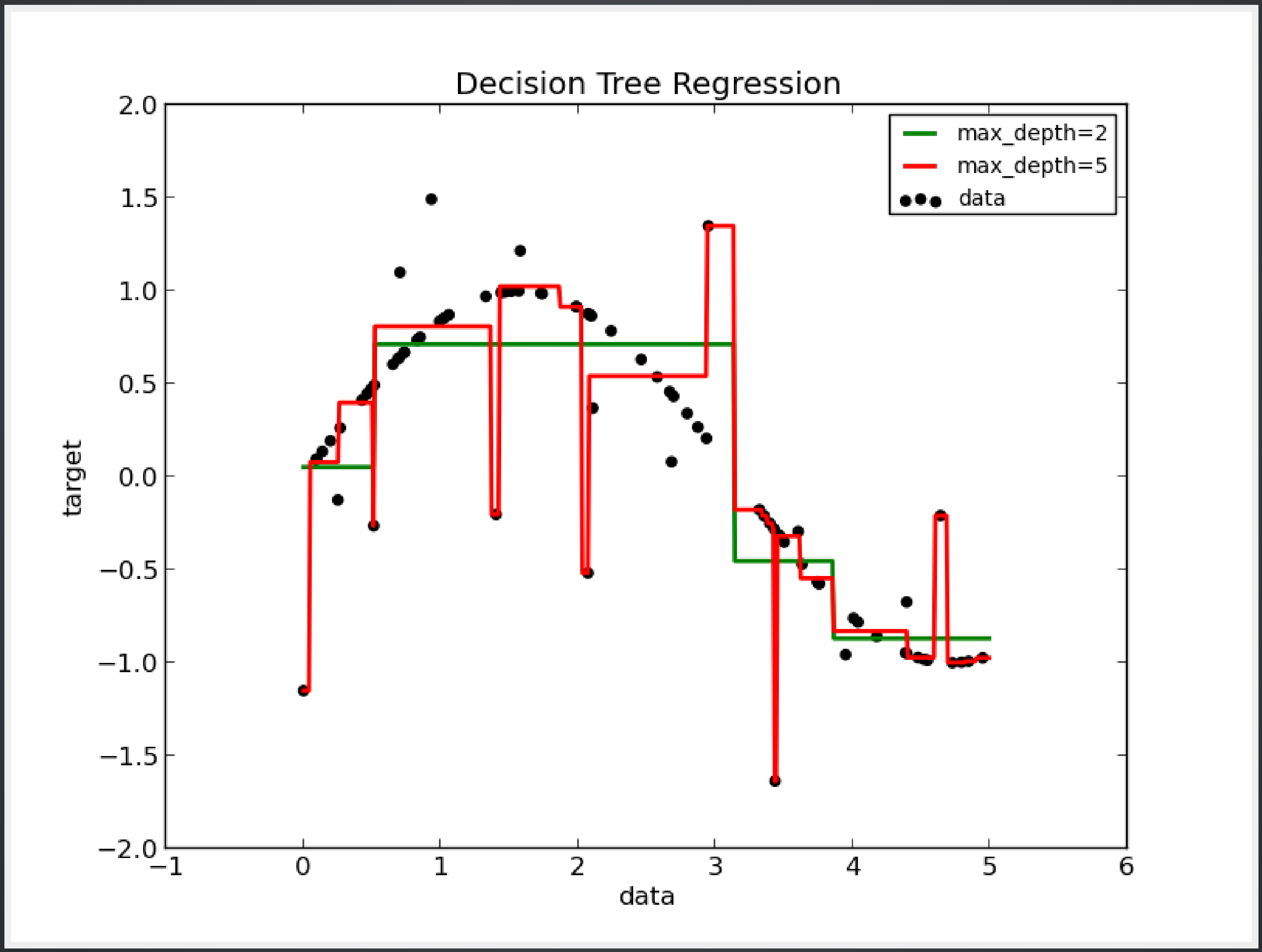
- Simple
- Does not need any distance or kernel

But

- tends to overfit
- nonoptimal decision rule
- produces very different classification rules
- has many variations
  (CART, ID3 and C4.5 standards)

REGRESSION WITH TREES

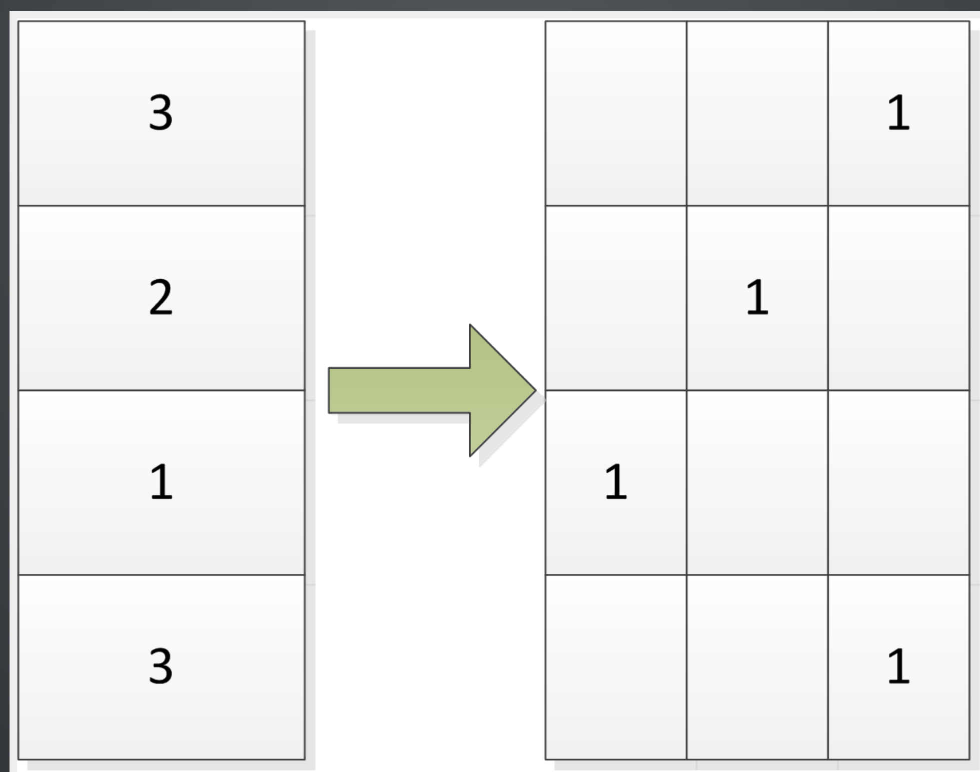
Uses greedy minimization of MSE / MAE



CATEGORICAL FEATURES

If there is no meaningful ordering in data (i.e. particle type, origin of particle, decay), tree and linear models will not be able to use this feature normally.

One-hot encoding trick:




THE END