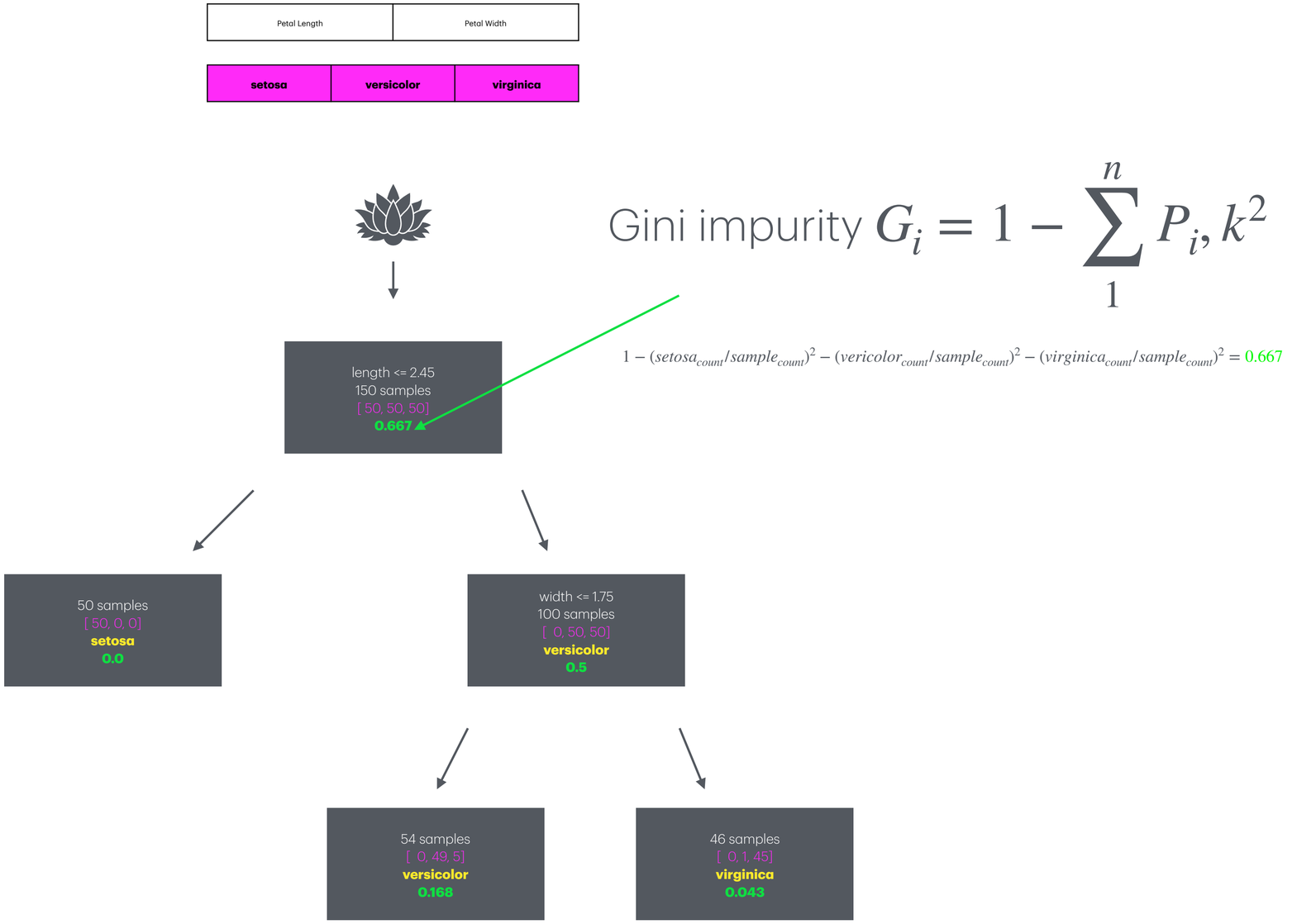
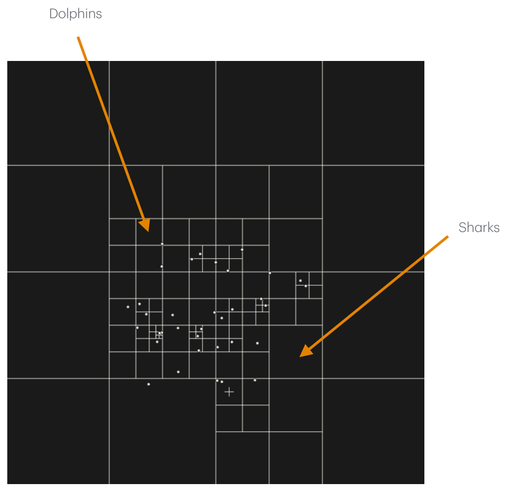
Decision Tree



Gini impurity $G_i = 1 - \sum_{1}^{n} P_i, k^2$

$1 - (setosa_{count}/sample_{count})^2 - (vericolor_{count}/sample_{count})^2 - (virginica_{count}/sample_{count})^2 = 0.667$

Goal is to keep branching until the impurity is low for all leaf nodes. The model works to create clustered region of low impurity regions



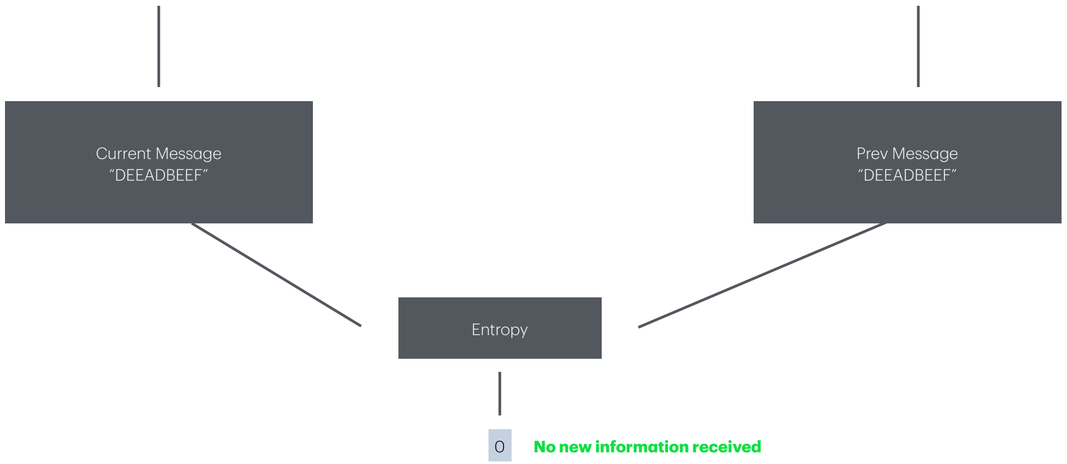
Entropy or Gini

Entropy is the measure of disorder(thermodynamics):

Zero entropy for constant (i.e. still) molecule and well ordered

Shannon Information Theory:

Entropy is average information content of message

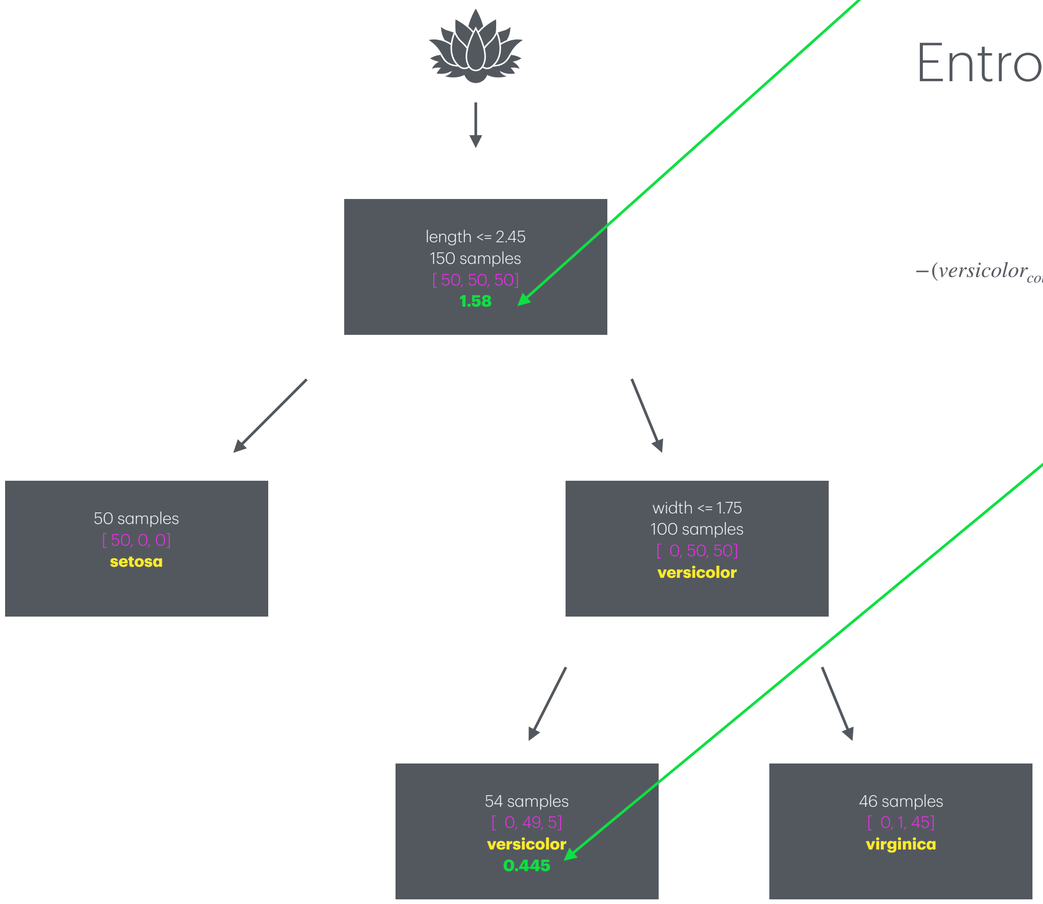


Machine Learning:

Entropy is zero when a set contains instances of only one class

$$-(setosa_{count}/sample_{count})log_2(setosa_{count}/sample_{count}) - (versicolor_{count}/sample_{count})log_2(versicolor_{count}/sample_{count}) - (virginica_{count}/sample_{count})log_2(virginica_{count}/sample_{count})$$

Entropy $H_i = -\sum_{k=1}^{n} P_i, k \; log_2(p_i, k)$



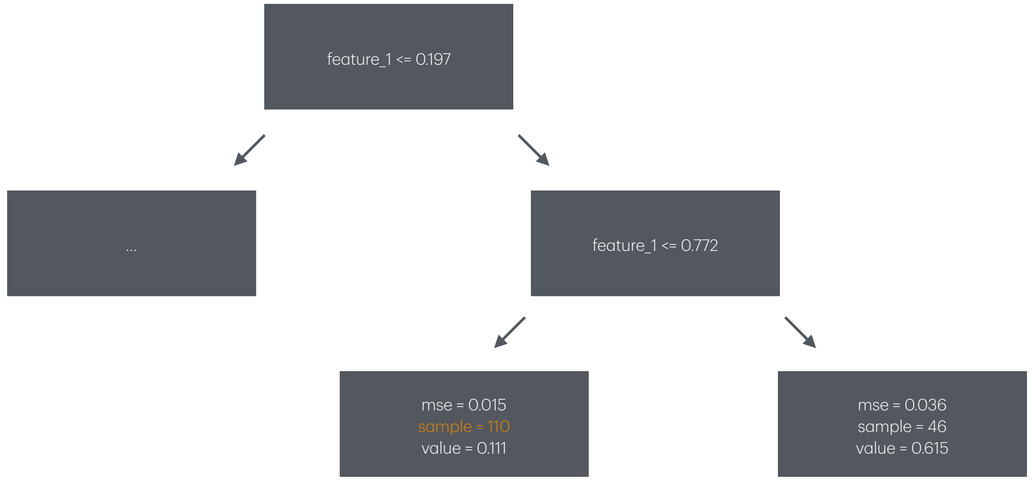
$$-(versicolor_{count}/sample_{count})log_2(versicolor_{count}/sample_{count}) - (virginica_{count}/sample_{count})log_2(virginica_{count}/sample_{count})$$

Non-parametric - Number of parameters is not determined prior to training.

Unlike parametric linear models, decision trees are infinite

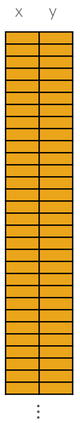
Decision trees are at risk to overfitting - restricting freedom is how decision tree models are regularized

Regression



CART cost function used to measure

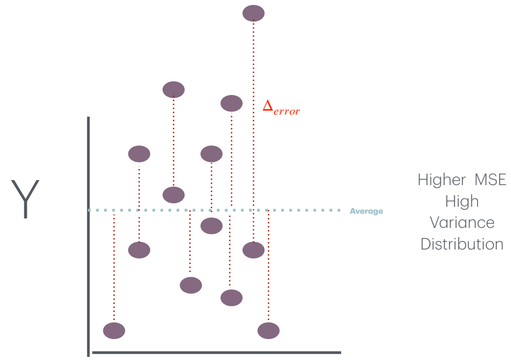
Training samples
feature_1 > 0.197 & feature_1 <= 0.772



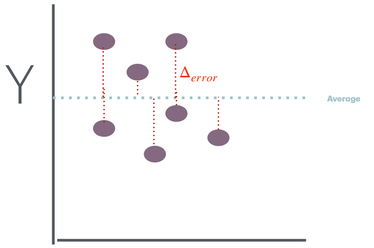
Prediction: $\hat{y}_{node} = \frac{1}{m_{node}} \sum_i y^{(i)}$

MSE Node: $\sum_i (\hat{y}_{node} - y^{(i)})^2$

MSE Node: $\sum_i (\Delta_{error})^2$



Higher MSE
High
Variance
Distribution



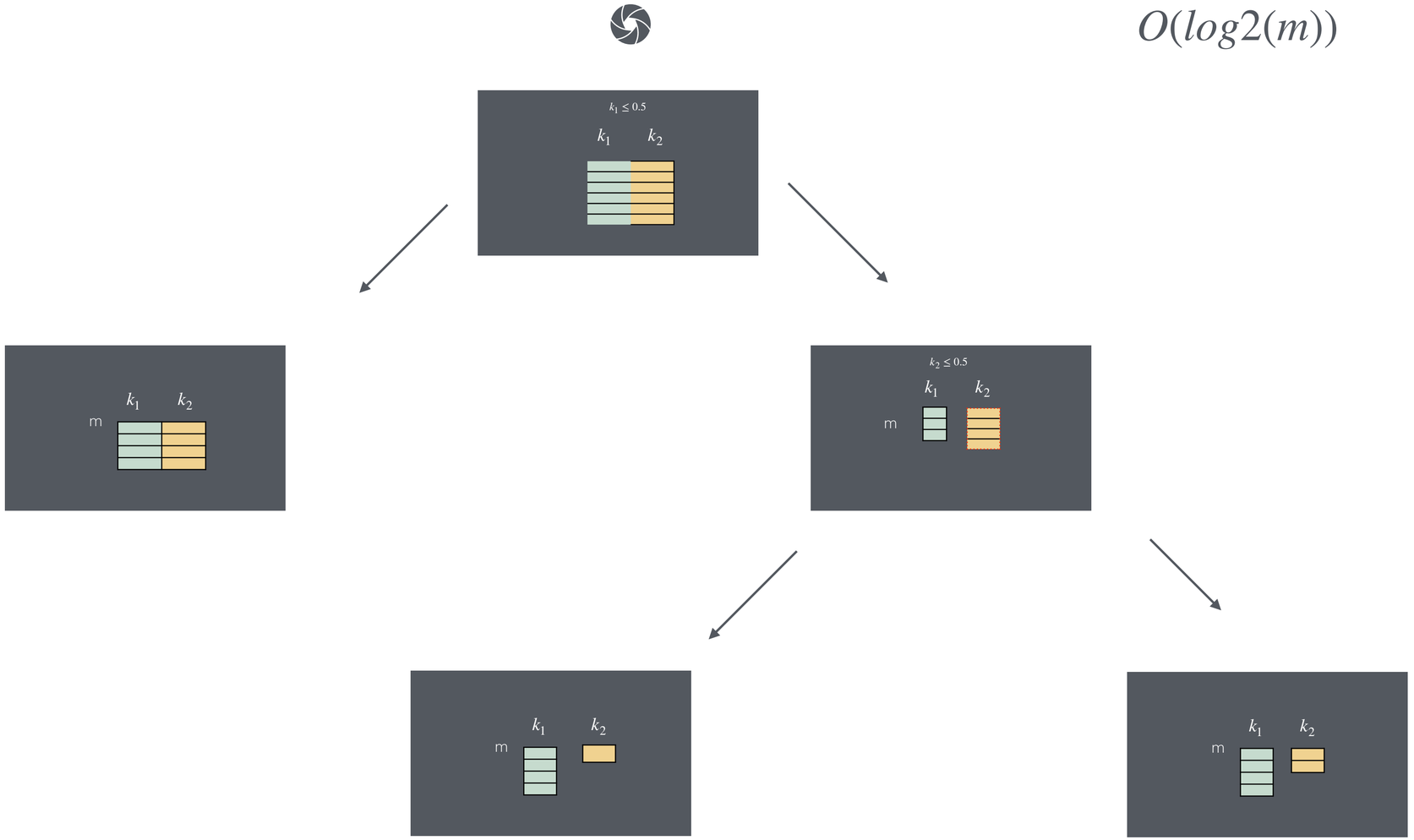
Lower MSE
Low
Variance
Distribution

MSE depends on the distribution of samples in node

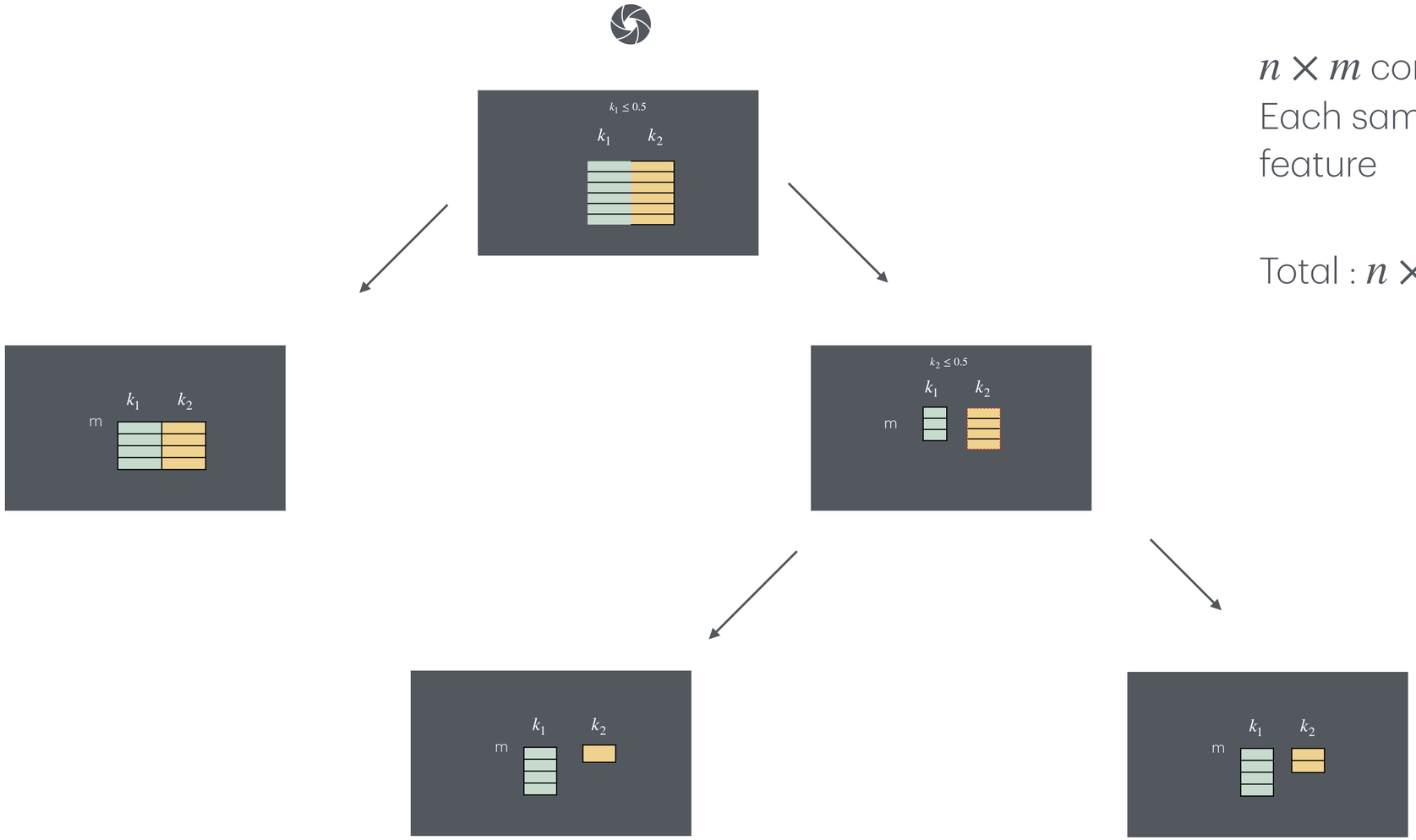
Prediction

Single check on each node is performed:

$$O(log2(m))$$



Training



Step to each node in tree fashion:

$O(log2(m))$

$n \times m$ compute at each node.
Each sample compares to each feature

Total : $n \times m\ O(log2(m))$

Training: Inside Node

Find optimal split pair (feature and threshold)

Create children nodes with reference to samples

⋮

Exercise 5

If it takes one hour to train a Decision Tree on a training set containing 1 million instances, roughly how much time will it take to train another Decision Tree on a training set containing 10 million instances

n- features in node

m- instances in node

$$n \cdot m \cdot log_2(m) = 1$$ **Calculate Z**

$$n \cdot m \cdot 10 \cdot log_2(m \cdot 10) = Z$$

$$\frac{n \cdot m \cdot 10 \cdot log_2(m \cdot 10)}{n \cdot m \cdot log_2(m)} = Z$$

$$\frac{10 \cdot log_2(m \cdot 10)}{log_2(m)} = Z$$

$$\frac{10 \cdot log(m \cdot 10)}{log(2)} \quad \frac{log(2)}{log(m)} = Z$$

$$\frac{10 \cdot log(m \cdot 10)}{log(m)} = Z$$ m= 1000000

$$11.67 = Z$$

Train Decision Tree

Preprocessing

↓

Split Training/Test

↓

Grid Search Find Optimal Decision Tree Classifier

↓

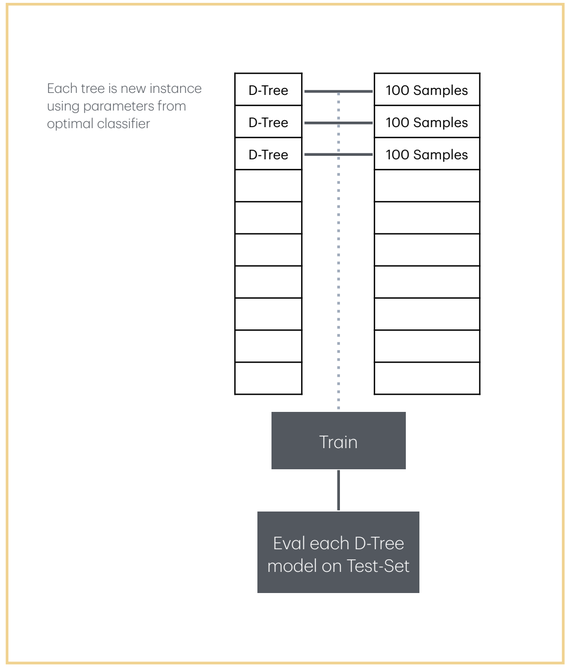
Generate 1000 Training subsets (size=100)

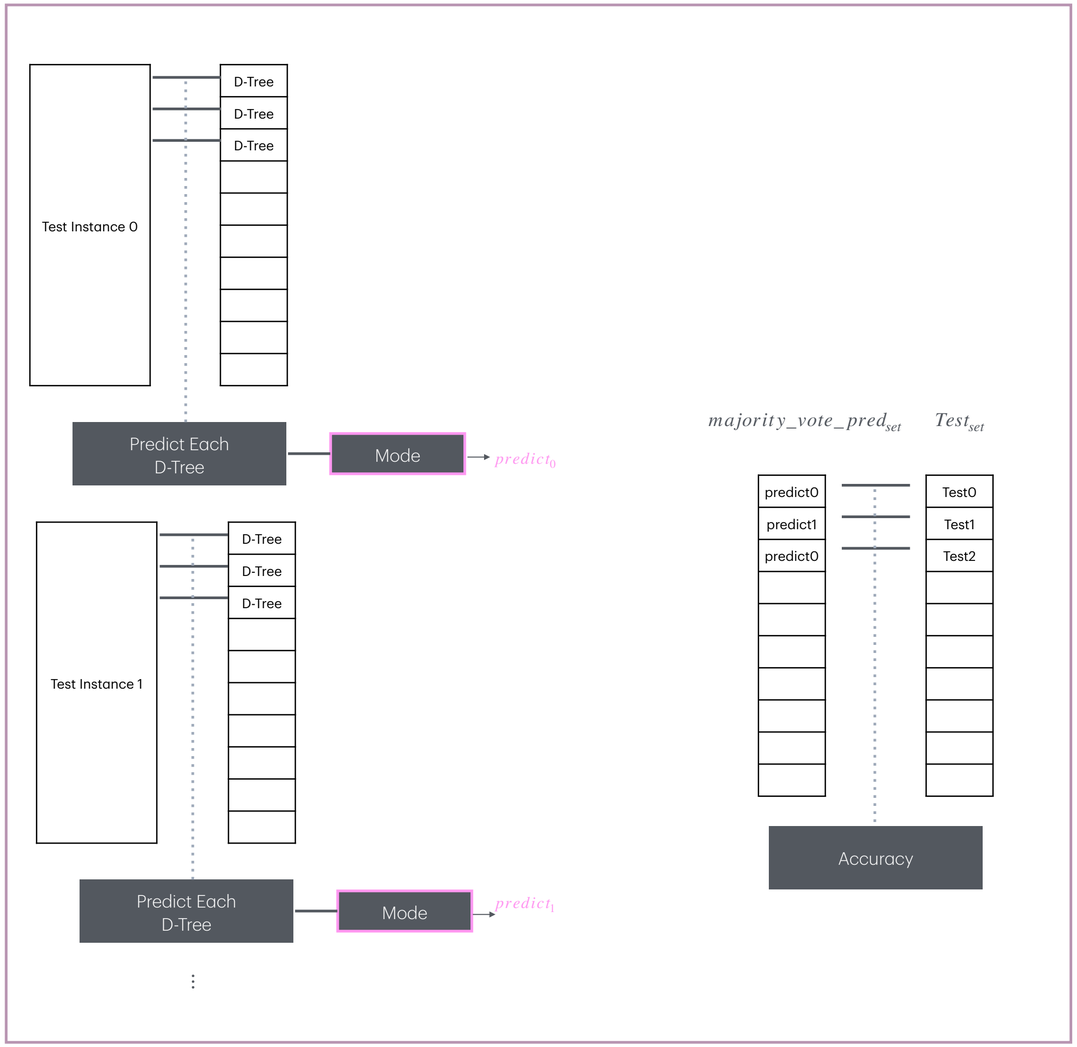
↓

Train 1000 D-Trees

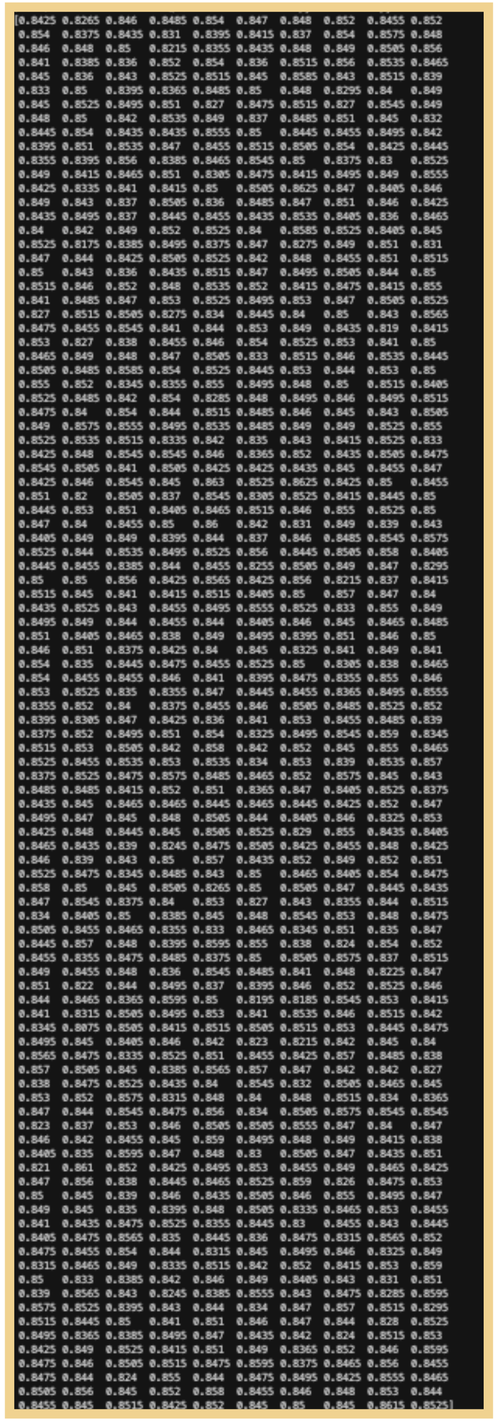
↓

Measure Majority





1000 D-Trees/
100 Sample
Predictions
Accuracy



Majority - Vote Predictions Accuracy

0.8575