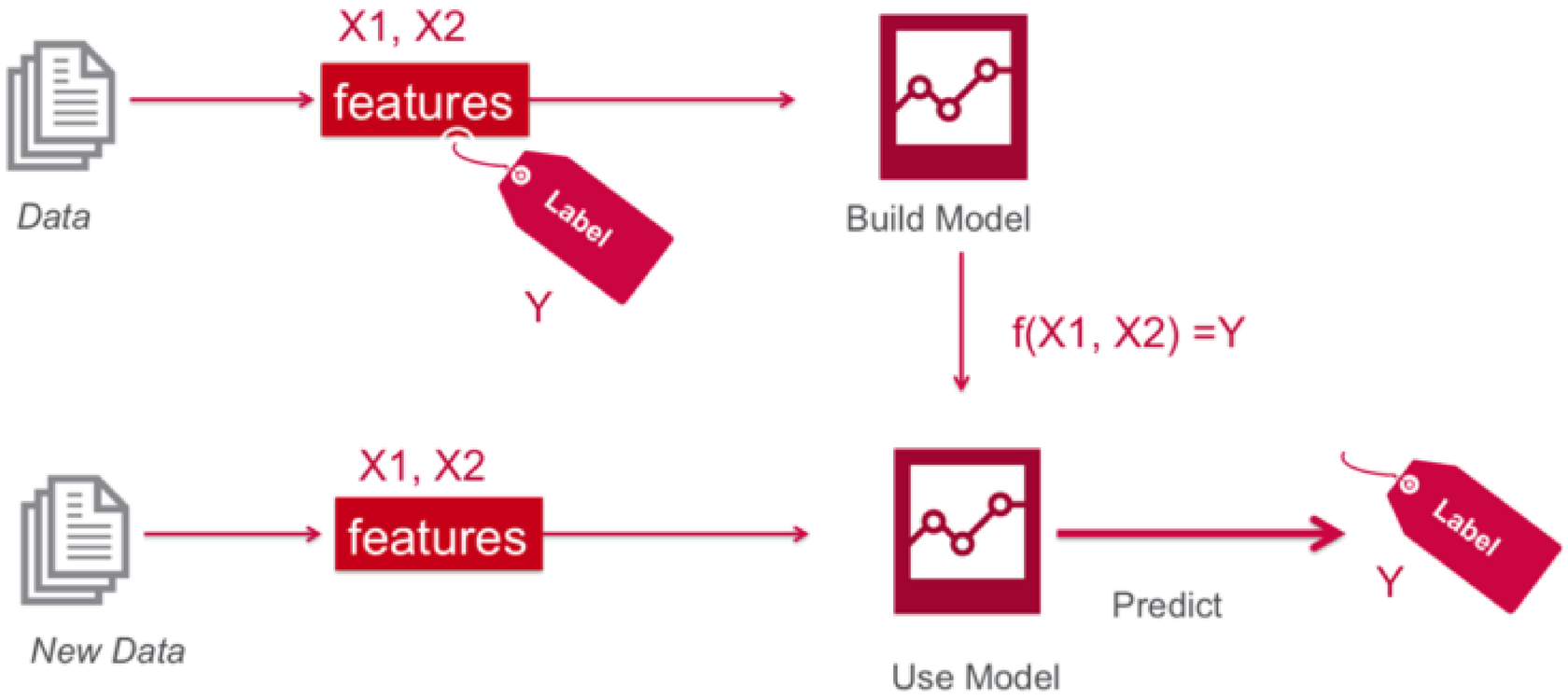
Decision tree
Classification

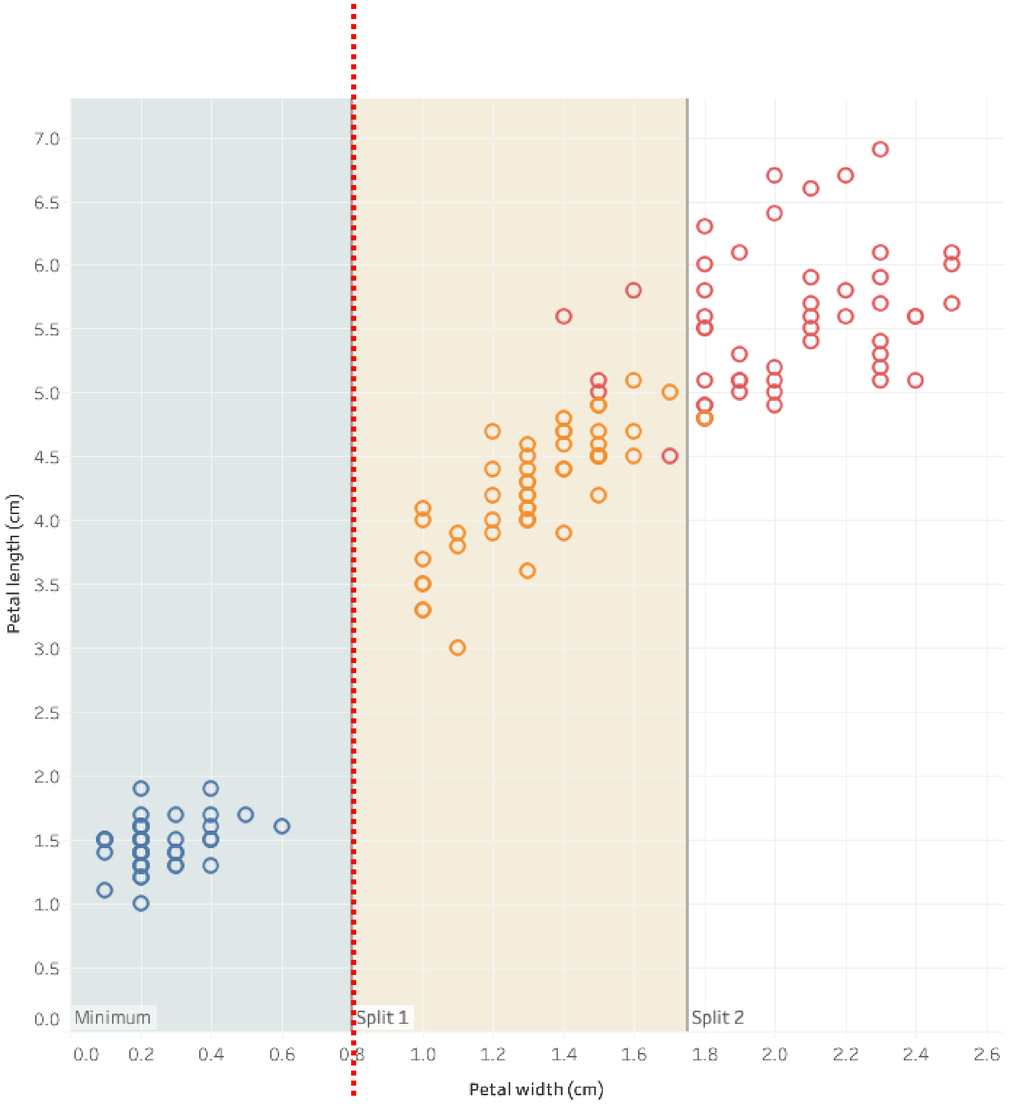
Supervised learning

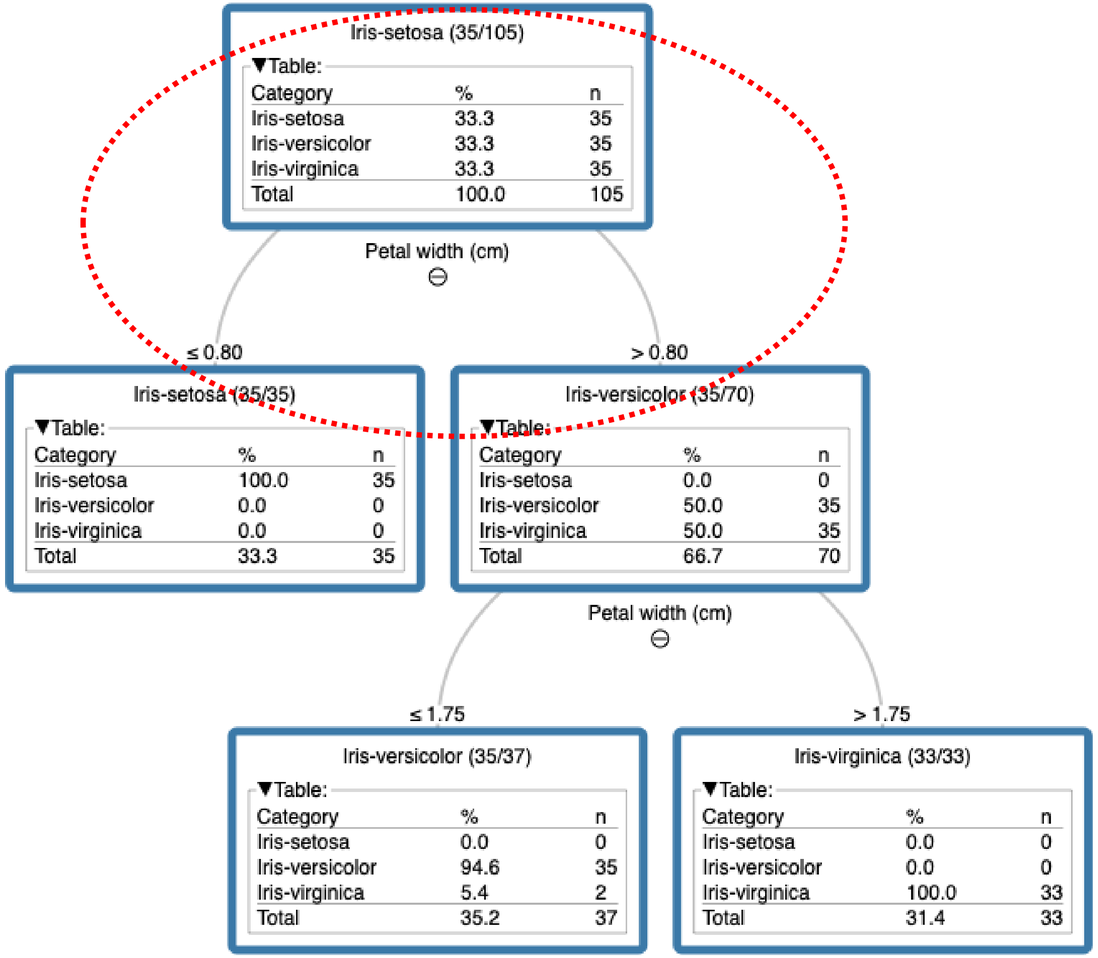


Demo in Tableau & KNIME

Back to our Iris example

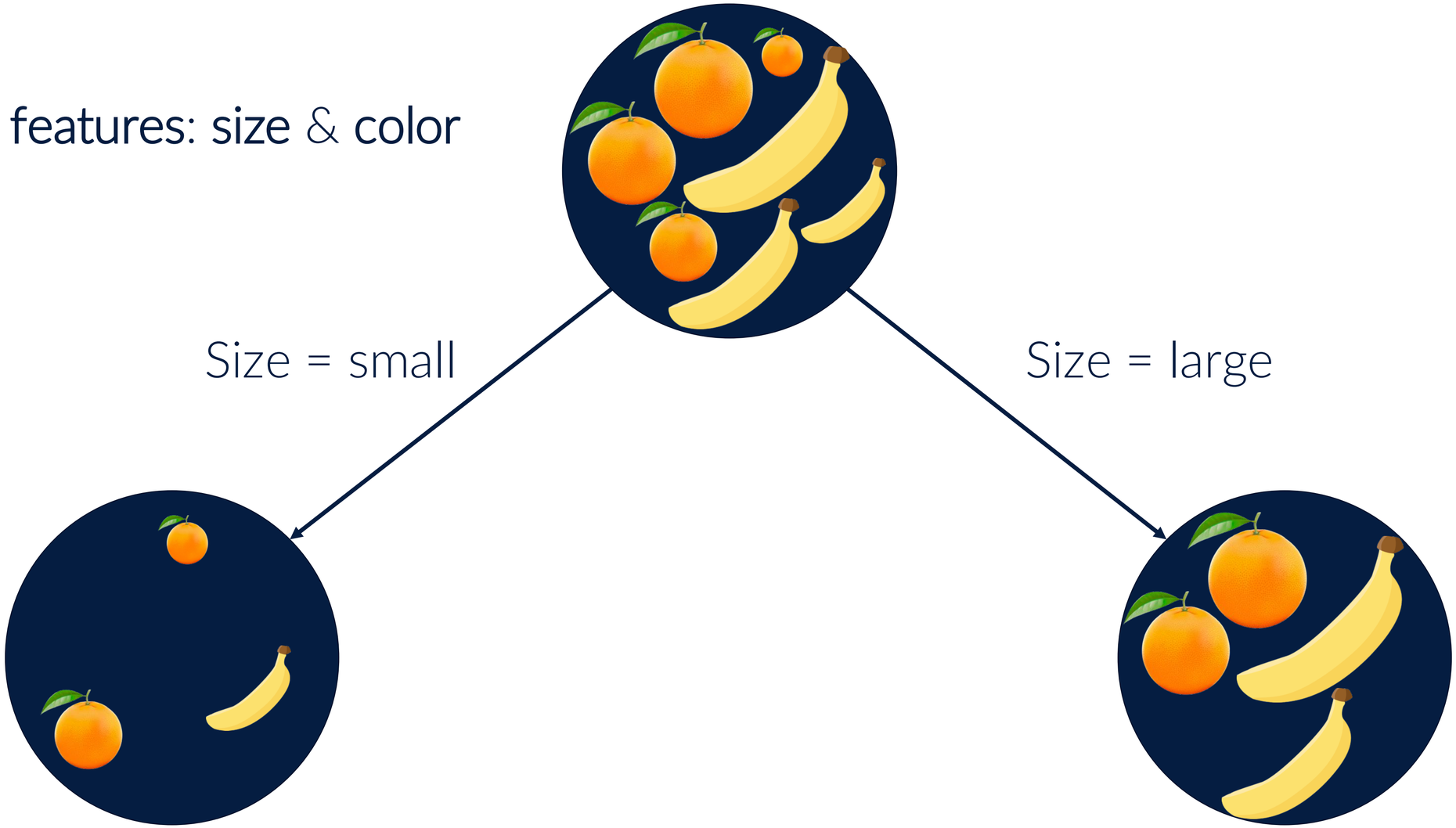
How is this first split decided?
Feature and split point of 0.8?





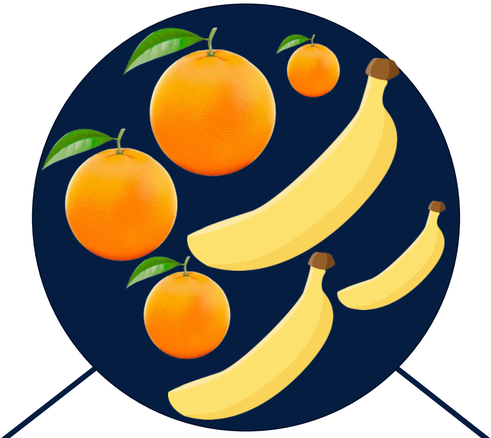
Measuring the quality of a split

Say, two features: size & color



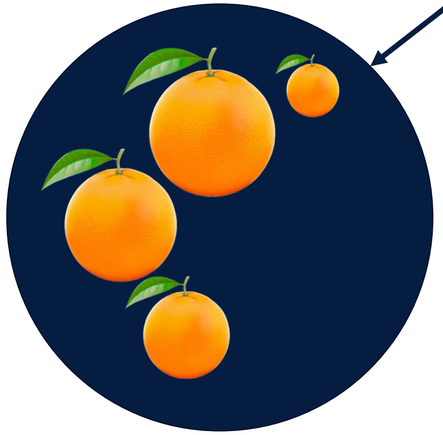
Measuring the quality of a split

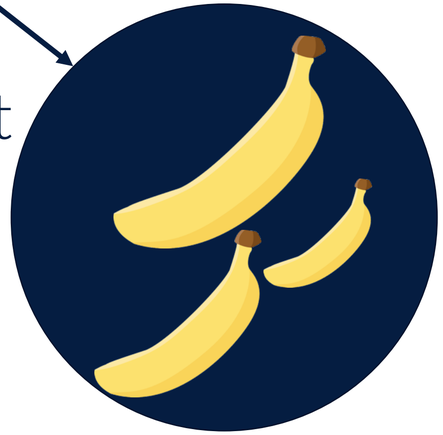
Say, **two features**: **size** & **color**



Color = orange

Color = yellow

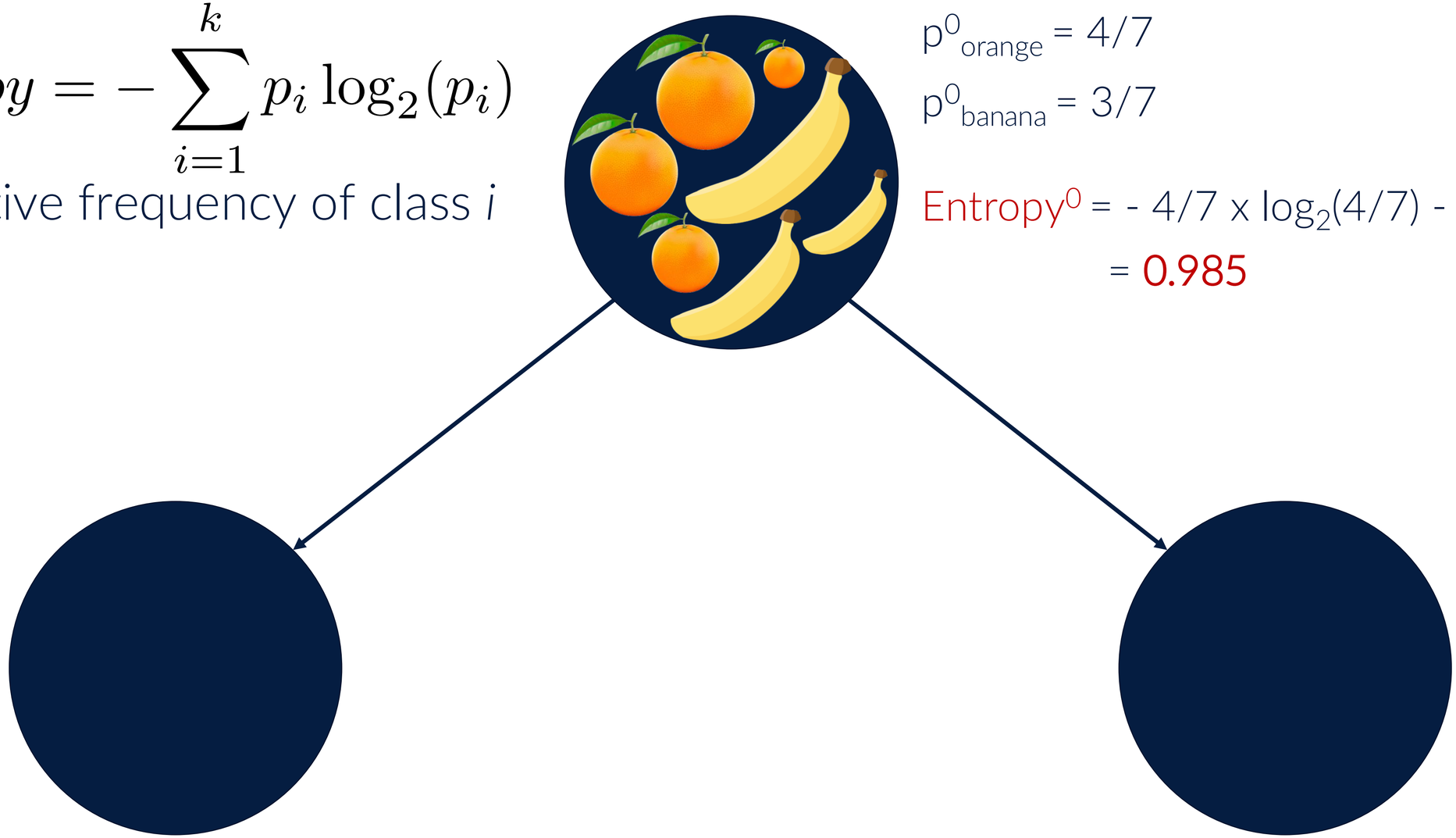




Pick the feature and split point that minimizes the **impurity** of the left & right children nodes

Entropy as a measure of impurity

- $Entropy = -\sum_{i=1}^{k} p_i \log_2(p_i)$
- p_i : relative frequency of class *i*



$p^0_{orange} = 4/7$

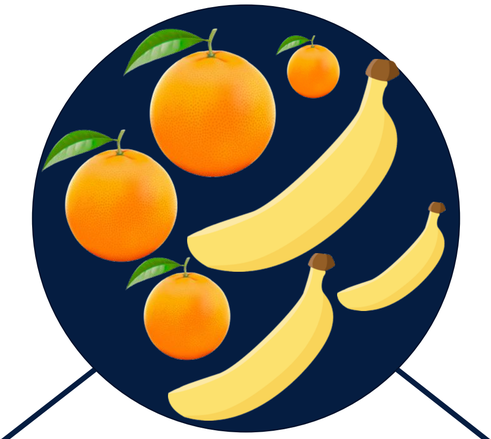
$p^0_{banana} = 3/7$

$Entropy^0 = -\ 4/7 \times \log_2(4/7) - 3/7 \times \log_2(3/7)$

$= 0.985$

Entropy as a measure of impurity

- $Entropy = -\sum_{i=1}^{k} p_i \log_2(p_i)$
- p_i : relative frequency of class *i*



$p^0_{orange} = 4/7$

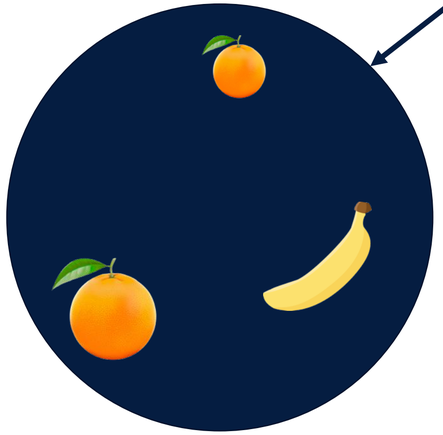
$p^0_{banana} = 3/7$

$Entropy^0 = -\ 4/7 \times \log_2(4/7) - 3/7 \times \log_2(3/7)$

$= 0.985$

Size = small

Size = large



$p^{1L}_{orange} = 2/3$

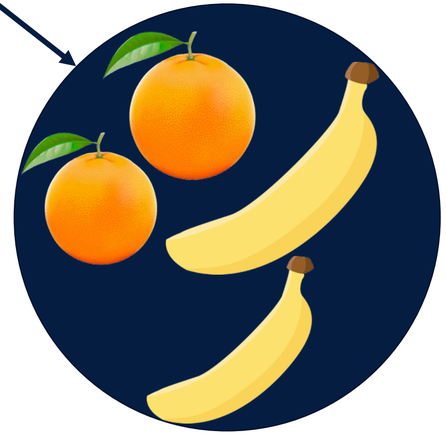
$p^{1L}_{banana} = 1/3$

$Entropy^{1L} = 0.918$

$p^{1R}_{orange} = 2/4$

$p^{1R}_{banana} = 2/4$

$Entropy^{1R} = 1.0$

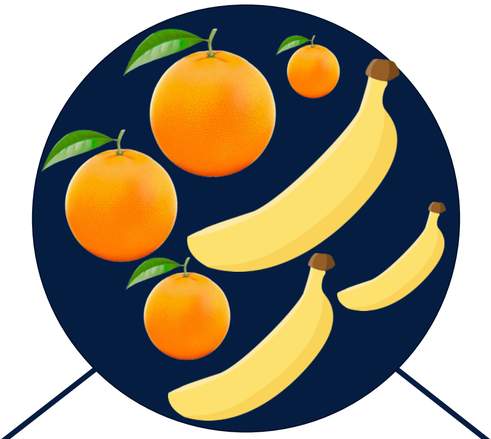


(Weighted) total entropy = 0.965

(3/7 x 0.918 + 4/7 x 1.0 = 0.965)

Entropy as a measure of impurity

- $Entropy = -\sum_{i=1}^{k} p_i \log_2(p_i)$
- p_i : relative frequency of class *i*



$p^0_{orange} = 4/7$

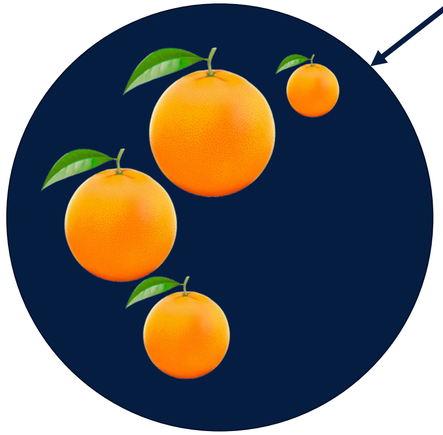
$p^0_{banana} = 3/7$

$Entropy^0 = -\ 4/7 \times \log_2(4/7) - 3/7 \times \log_2(3/7)$

$= 0.985$

Color = orange

Color = yellow



$p^{1L}_{orange} = 4/4$

$p^{1L}_{banana} = 0/4$

$Entropy^{1L} = 0.0$

$p^{1R}_{orange} = 0/4$

$p^{1R}_{banana} = 4/4$

$Entropy^{1R} = 0.0$

(Weighted) total entropy = 0.0 !

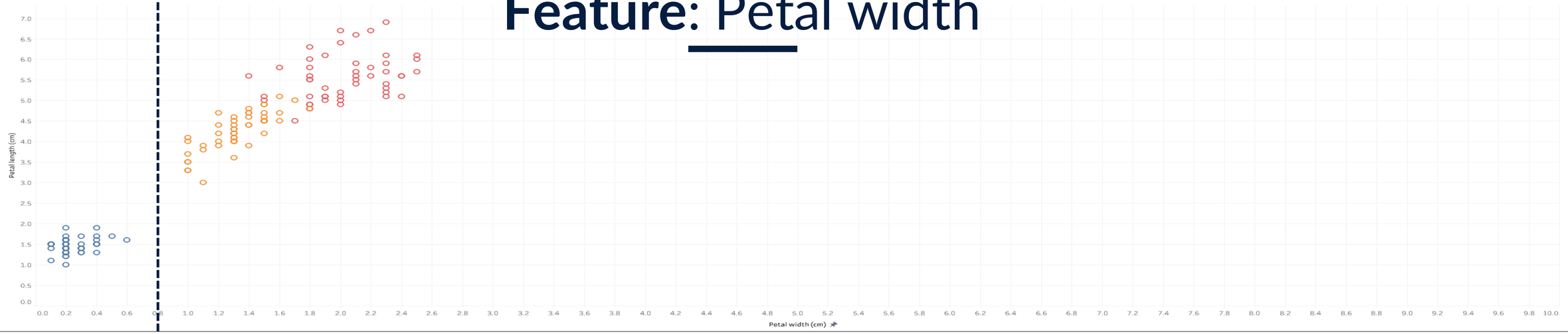
Information Gain as split decision

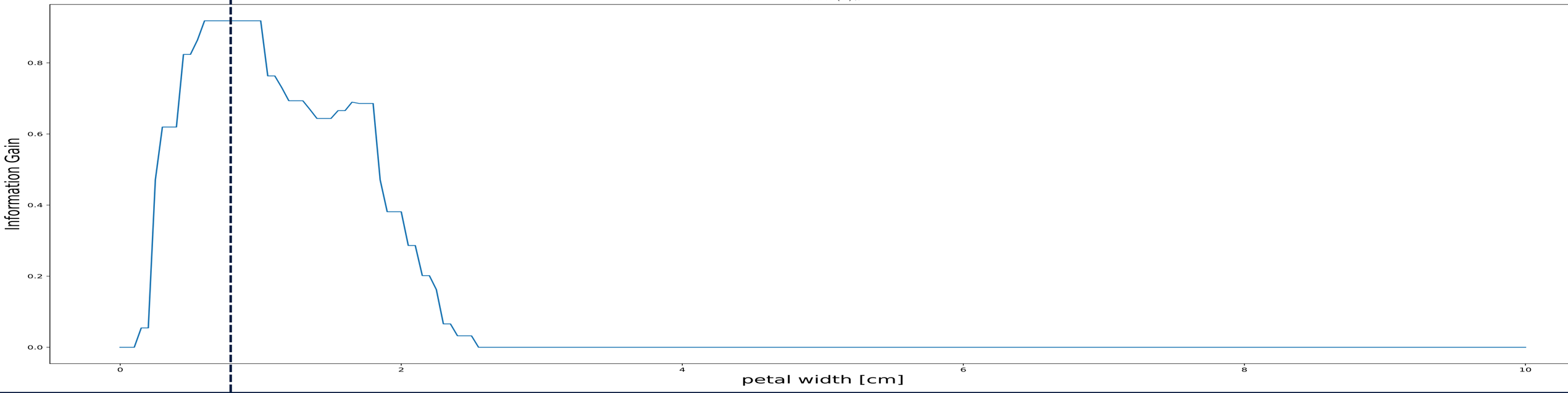
- Information Gain = How much entropy we removed by the split
- The split (feature & value) that maximizes the Information Gain is chosen

```
Split by size:     0.985 - 0.965 = 0.02
Split by color:    0.985 - 0.0   = 0.985
```

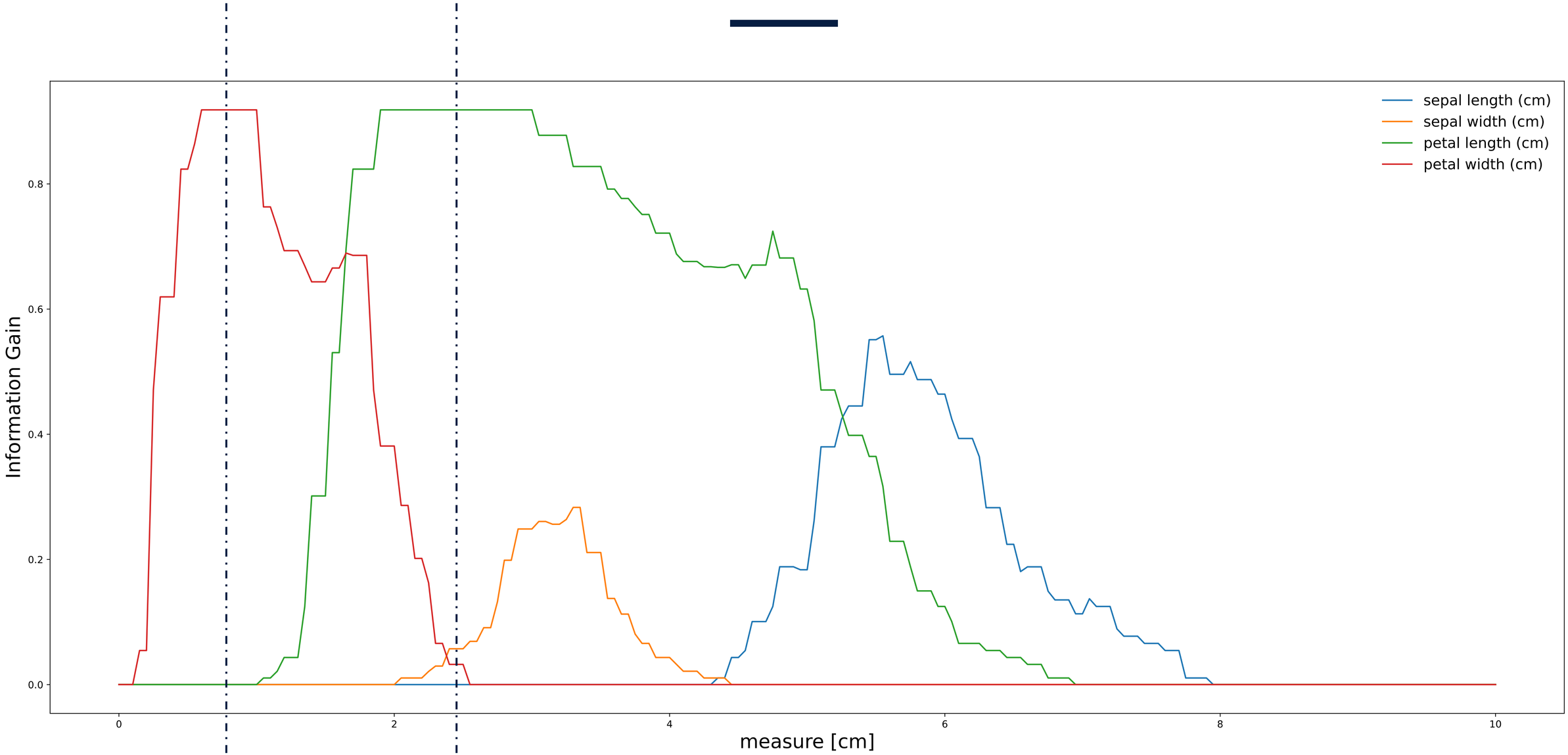
Back to our Iris example

Feature: Petal width





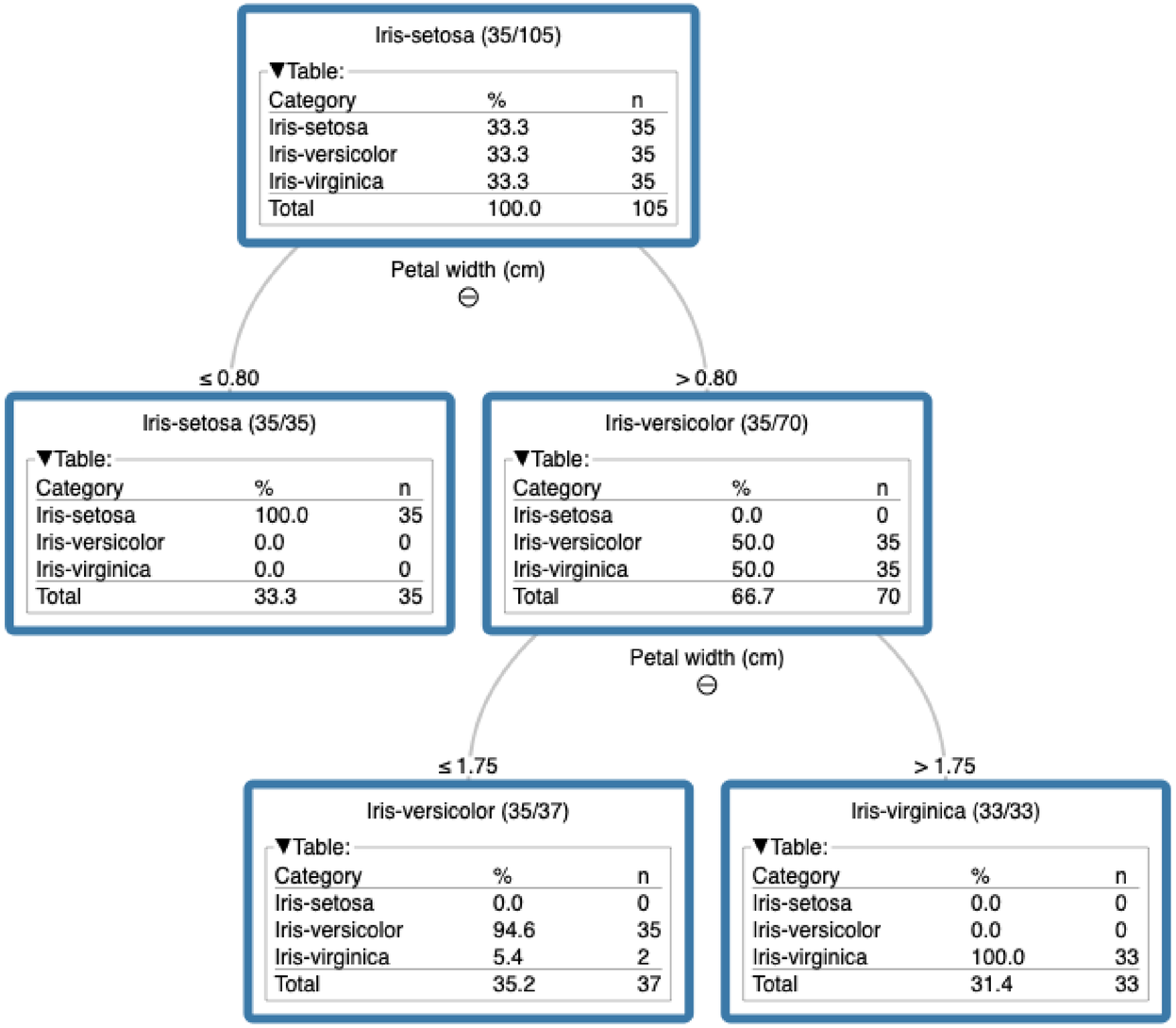
Over all 4 features



Recursively splitting

Stopping criterion (overfitting!)

- purity > *threshold*
- number of samples < *threshold*
- depth > *threshold*



Another split criterion

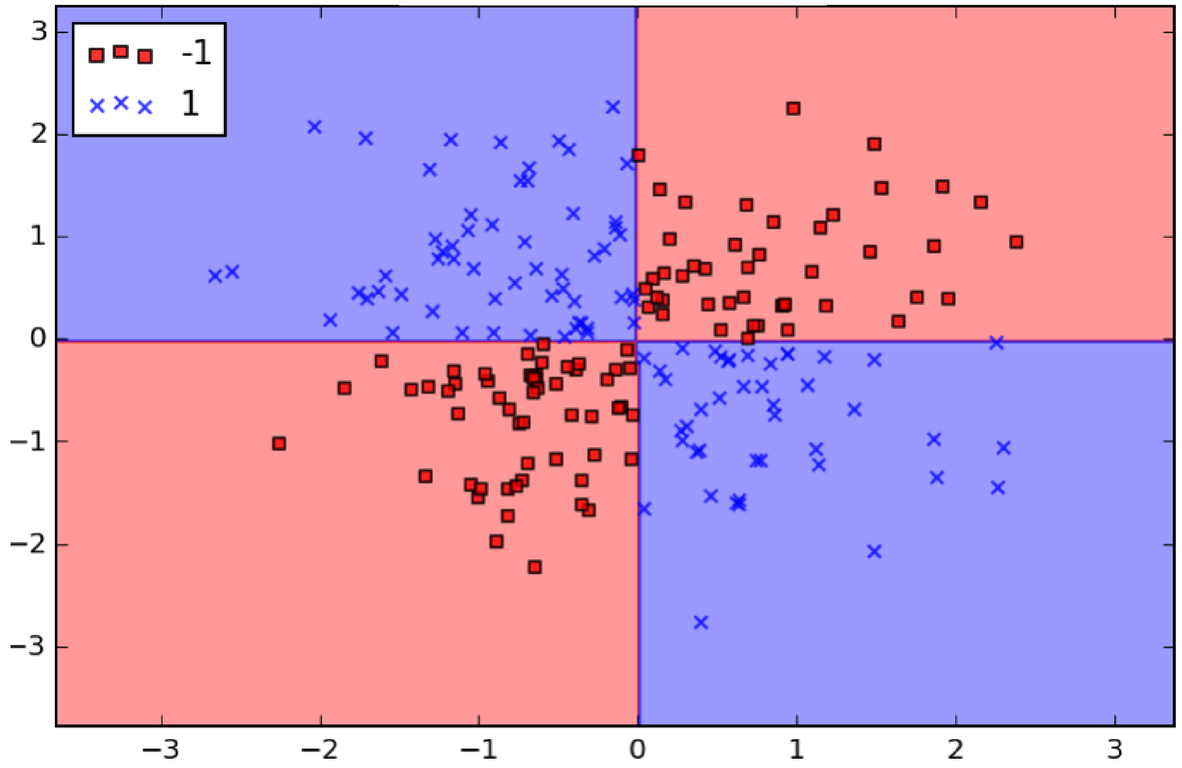
Gini index (or Gini impurity)

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

- The Gini Impurity of a dataset is a number between 0 and 0.5
- Gini impurity measures how often a randomly chosen element of a set (of cardinality k) would be incorrectly labeled if it were labeled randomly and independently according to the distribution of labels in the set.

More complex datasets

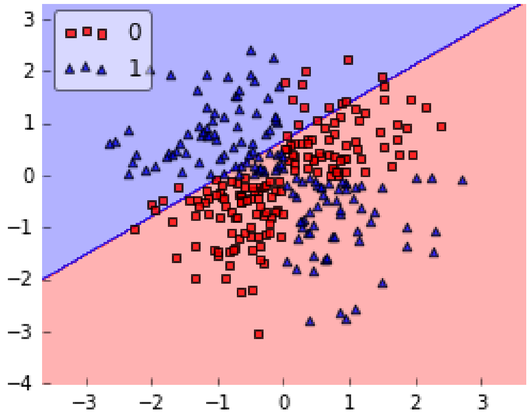
XOR



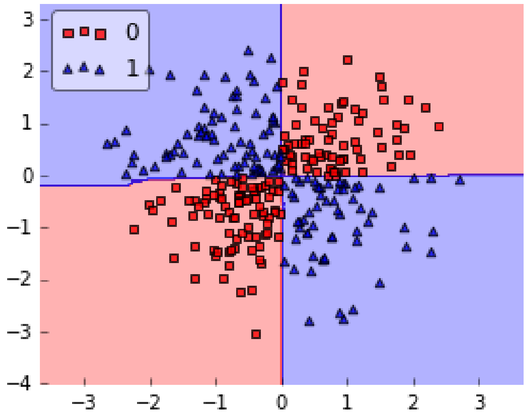
More complex datasets

XOR

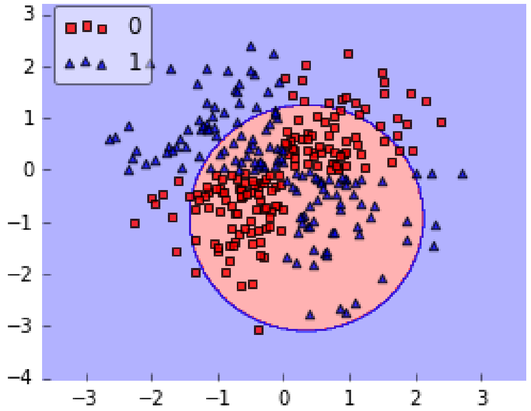
Logistic regression



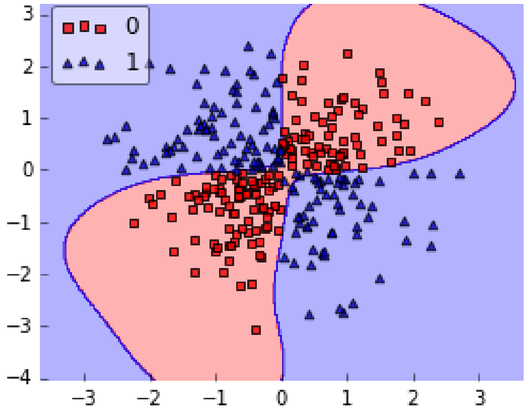
Deeper decision tree



Naïve Bayes

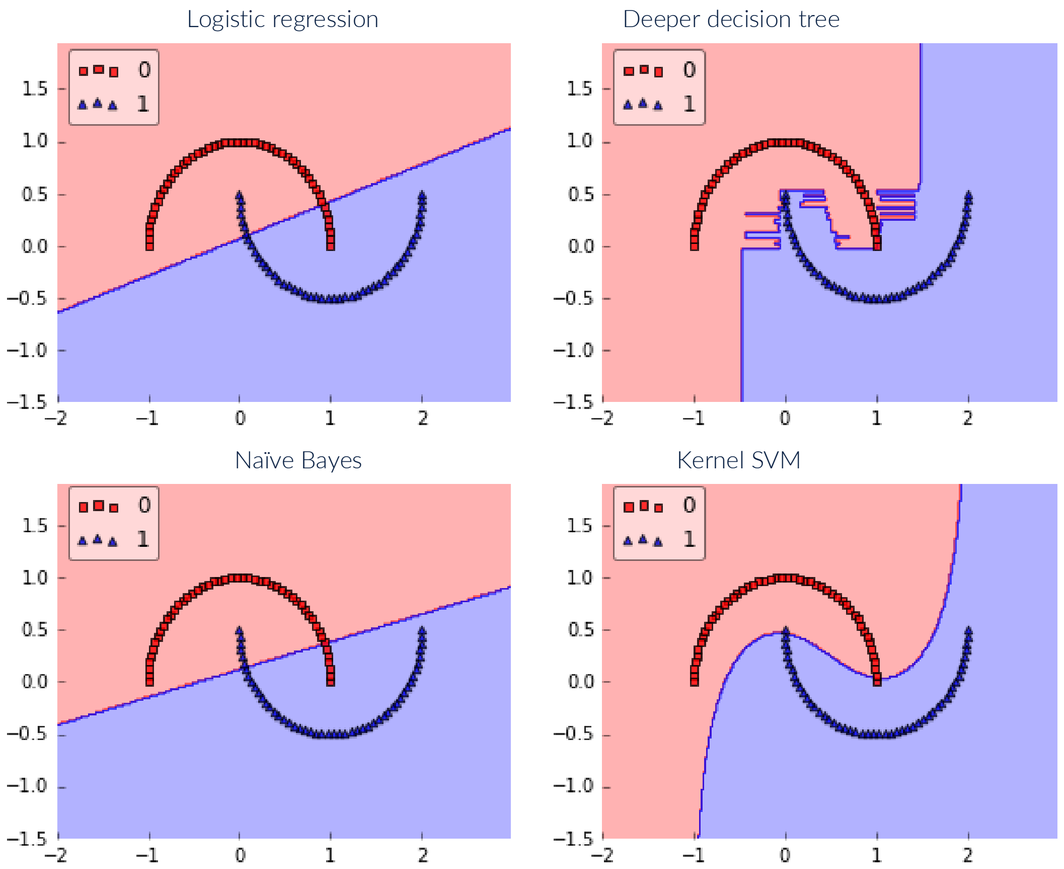


Kernel SVM



More complex datasets

Half-Moons



(Dis)Advantages of Decision Trees

Advantages:

1. Interpretability
2. Less Data Preparation
3. Non-Parametric
4. Non-Linearity

Disadvantages:

1. Overfitting
2. Unstability

Random forest classifier

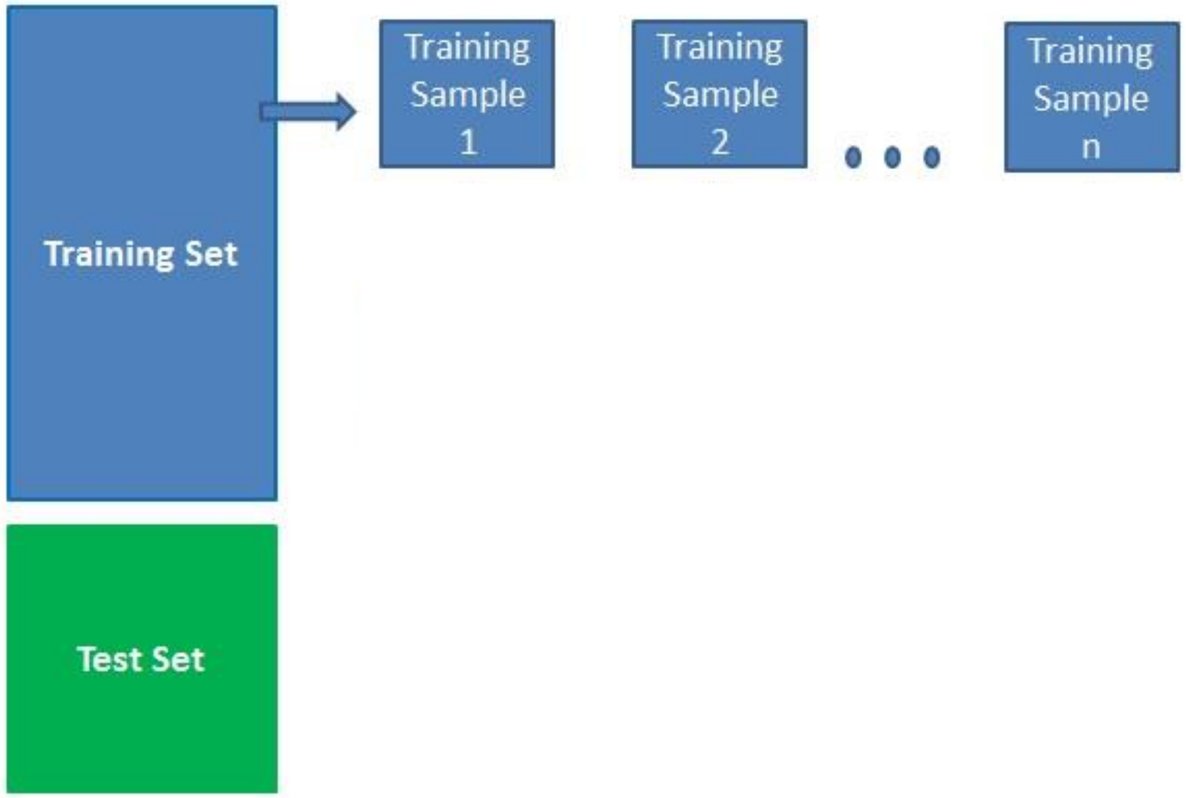
- More stable
- Less prone to overfitting

Random forest classifier

- Works in 4 steps

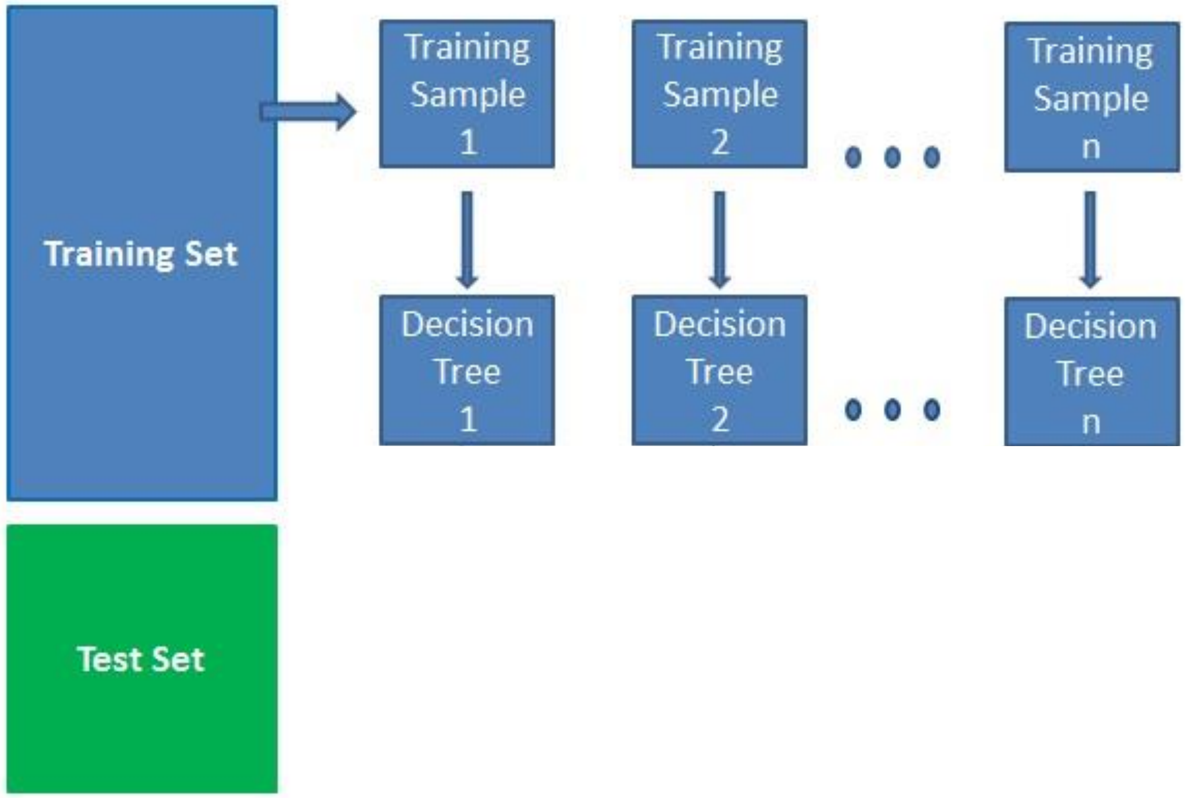
Random forest classifier

- Works in 4 steps:
  1. Select random samples from a given dataset



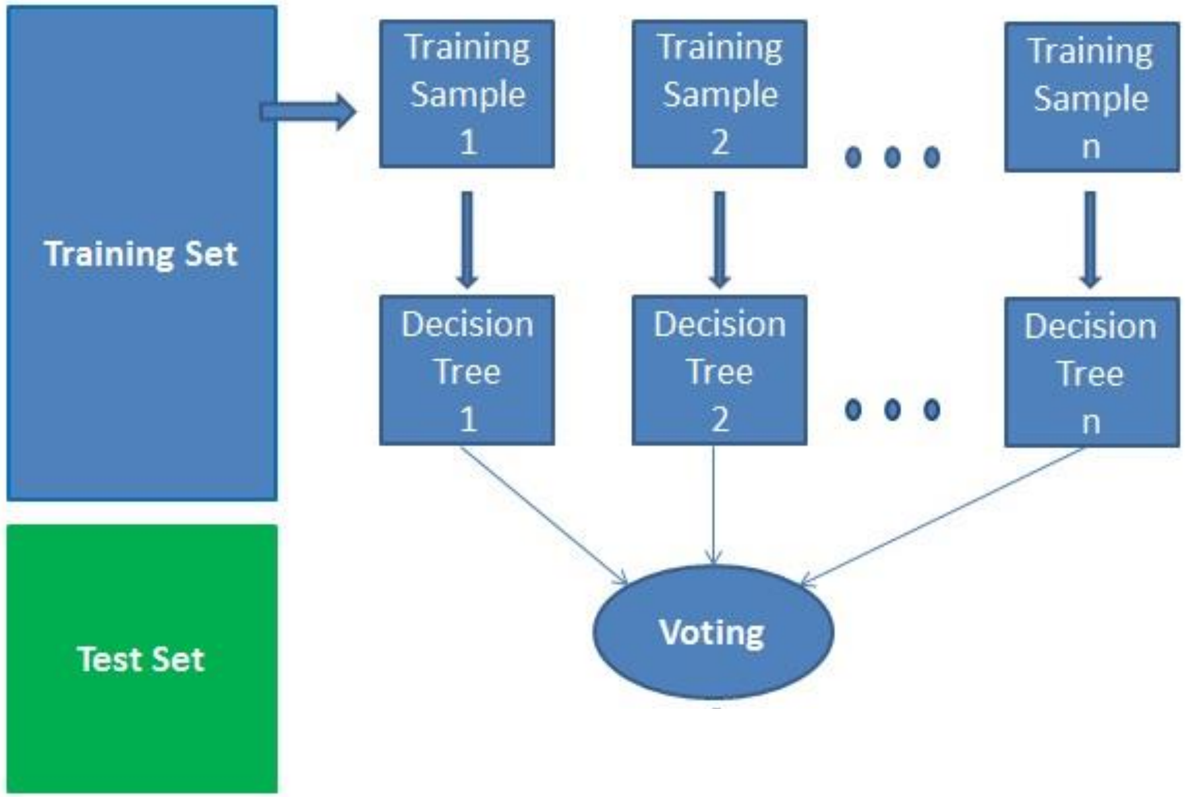
Random forest classifier

- Works in 5 steps:
  1. Select random samples from a given dataset
  2. Construct a decision tree for each sample



Random forest classifier

- Works in 5 steps:
  1. Select random samples from a given dataset
  2. Construct a decision tree for each sample
  3. Get a prediction from each tree
  4. Perform a vote for each predicted result



Random forest classifier

- Works in 5 steps:
  1. Select random samples from a given dataset
  2. Construct a decision tree for each sample
  3. Get a prediction from each tree
  4. Perform a vote for each predicted result
  5. Select the prediction result with the most votes as the final prediction

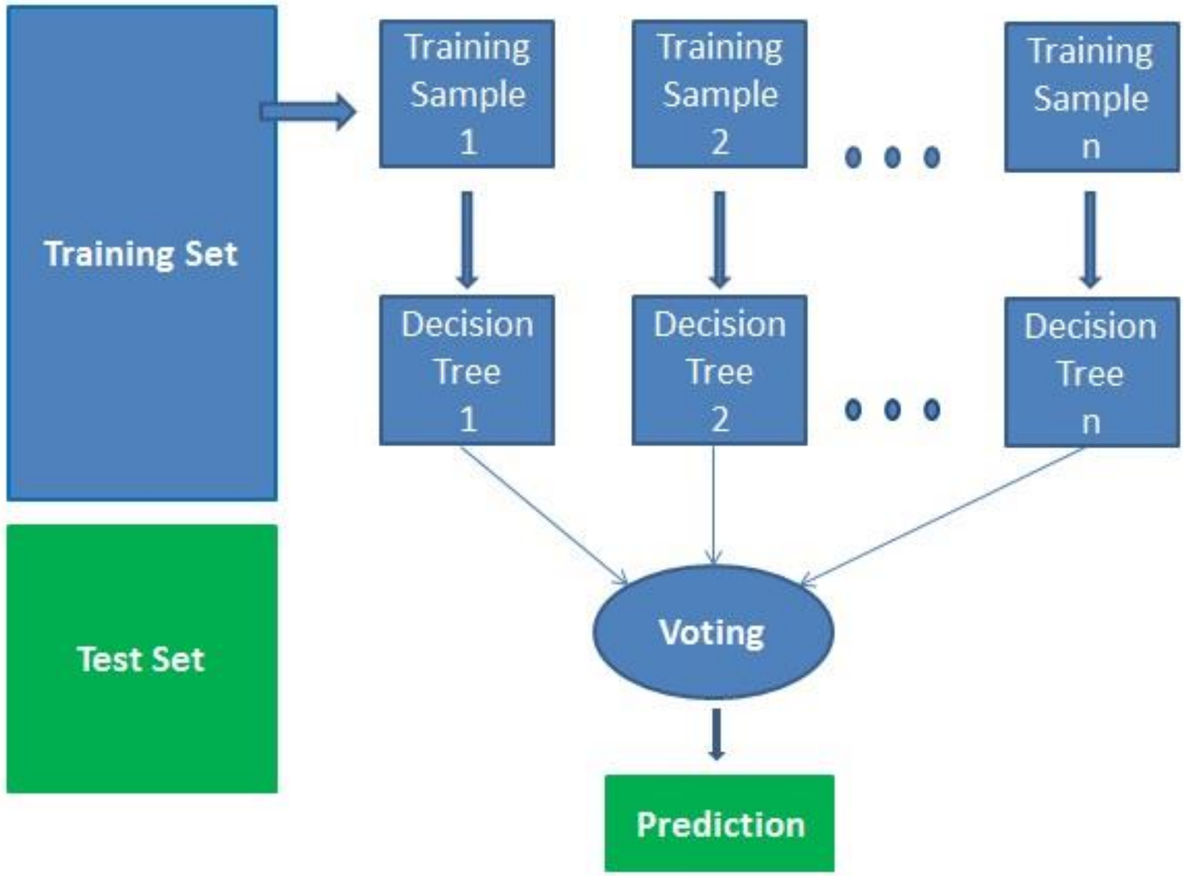
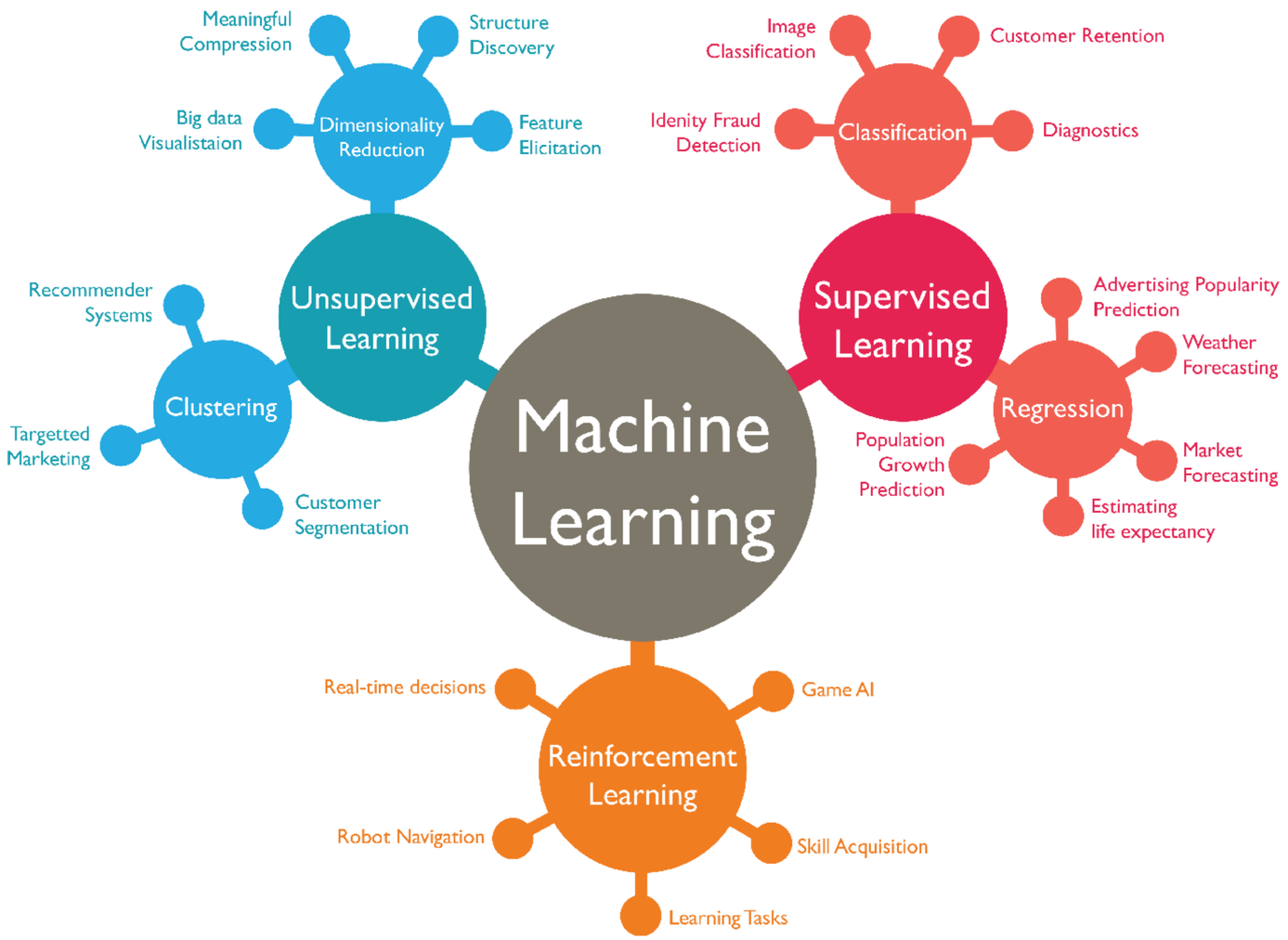
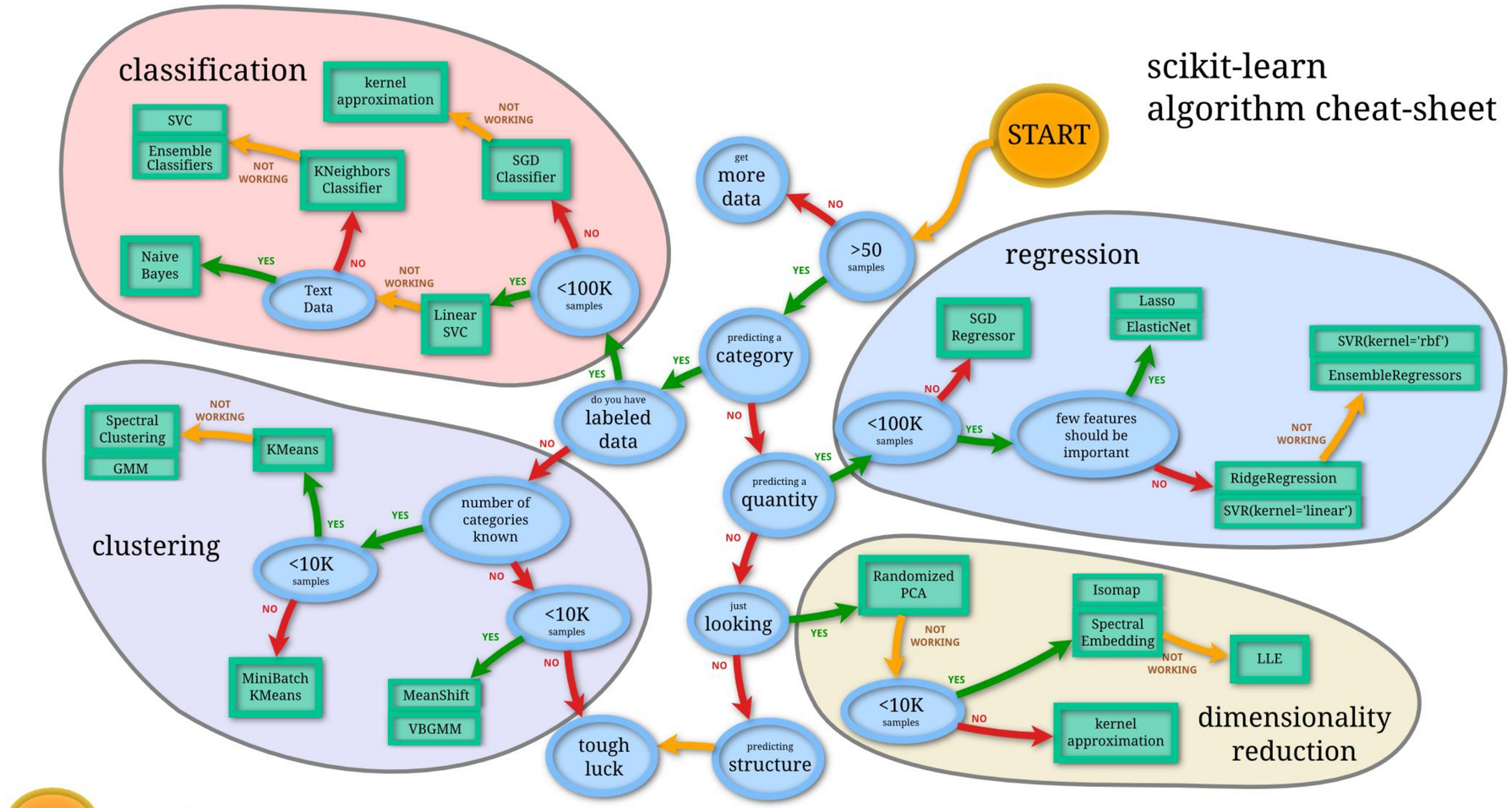


Illustration in Python
(cf notebook)

Types of learning

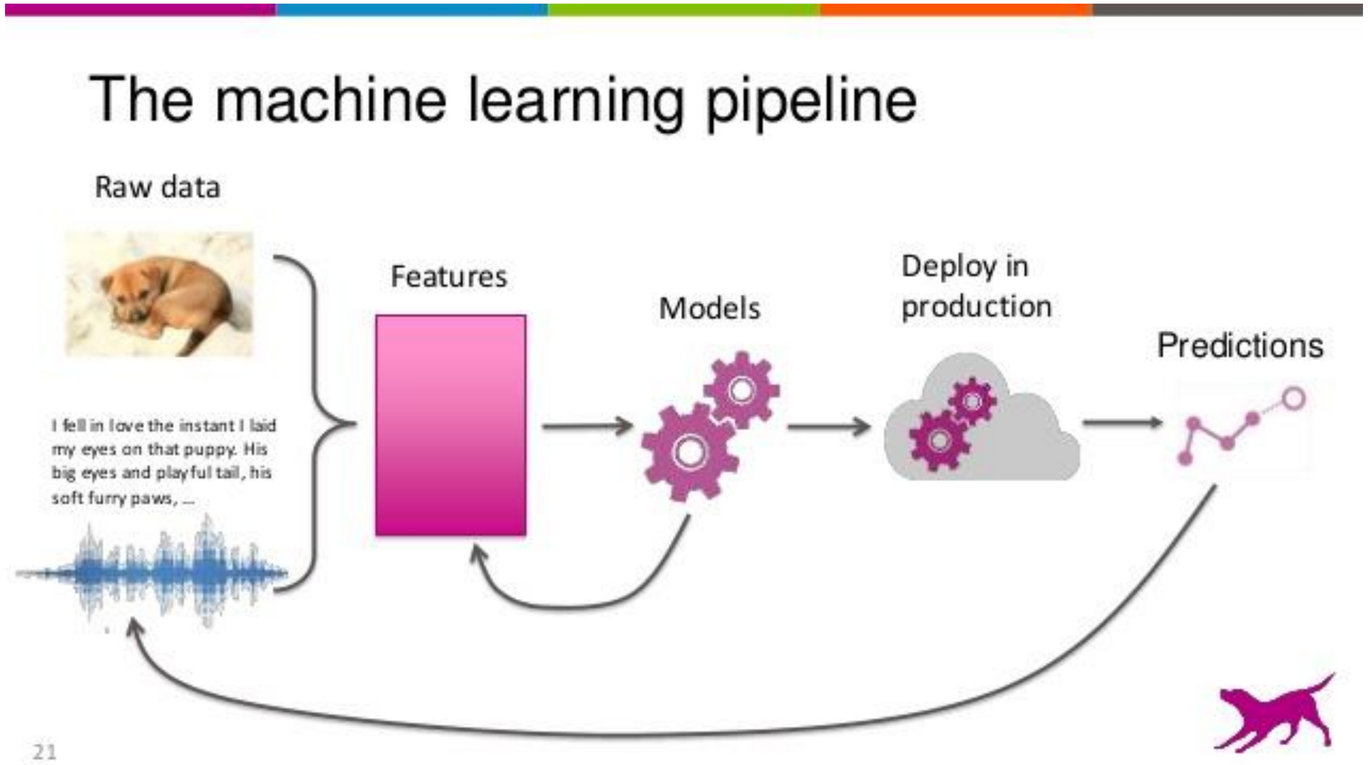


Choosing the right model



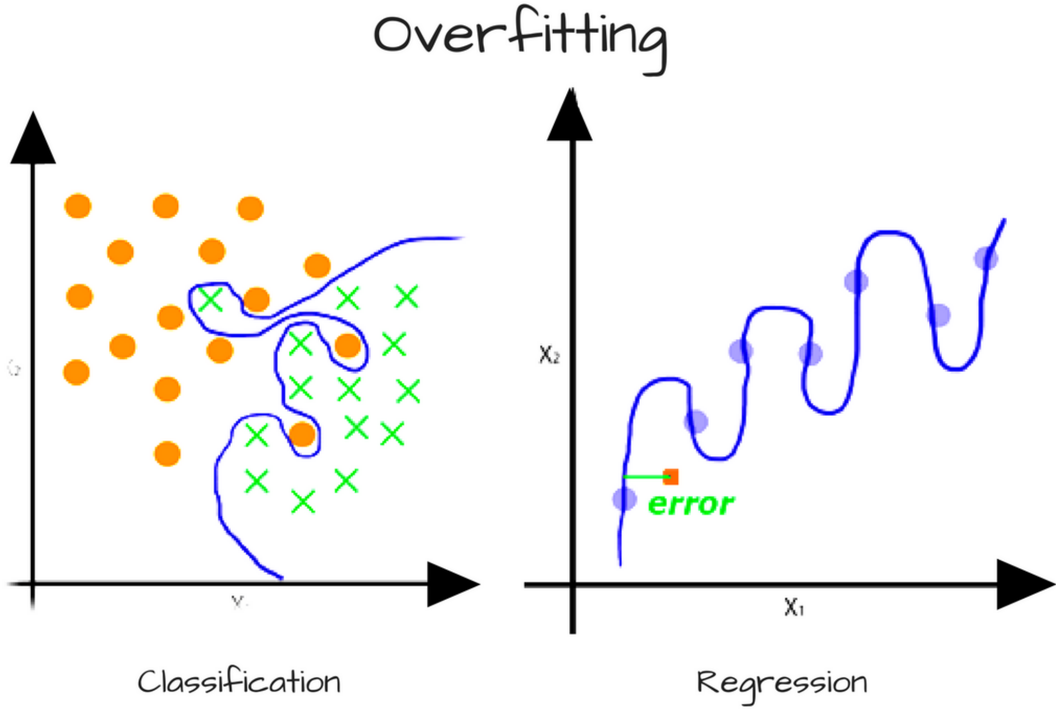
Final (important) words

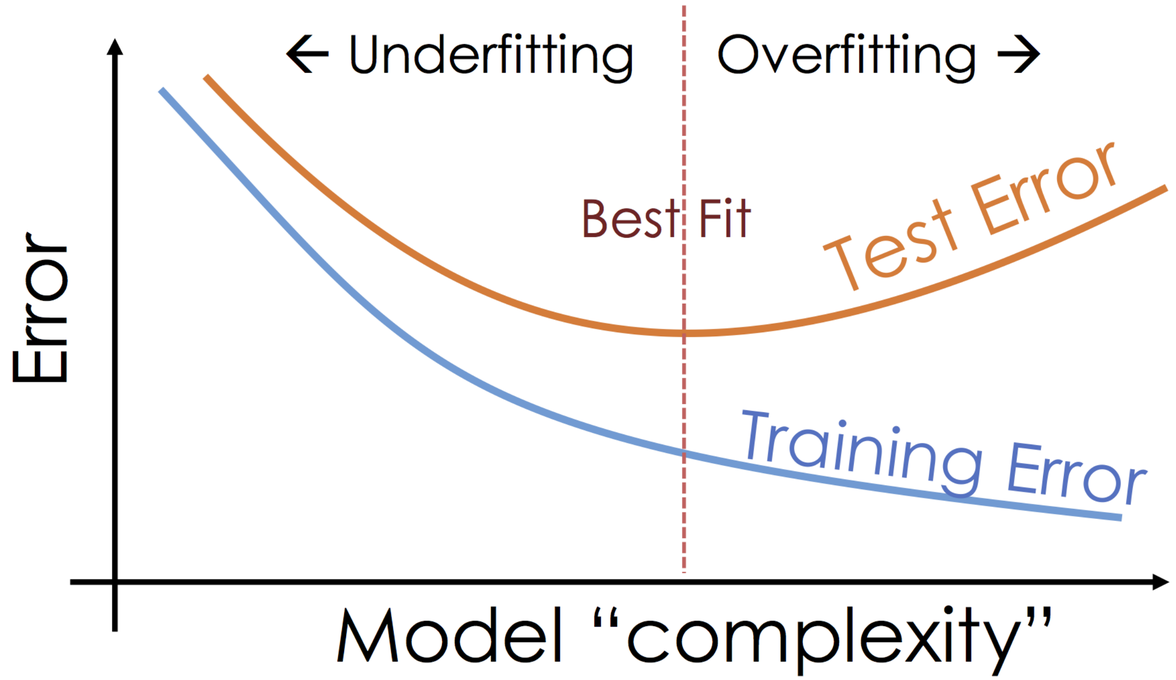
Feature engineering & selection



Final (important) words

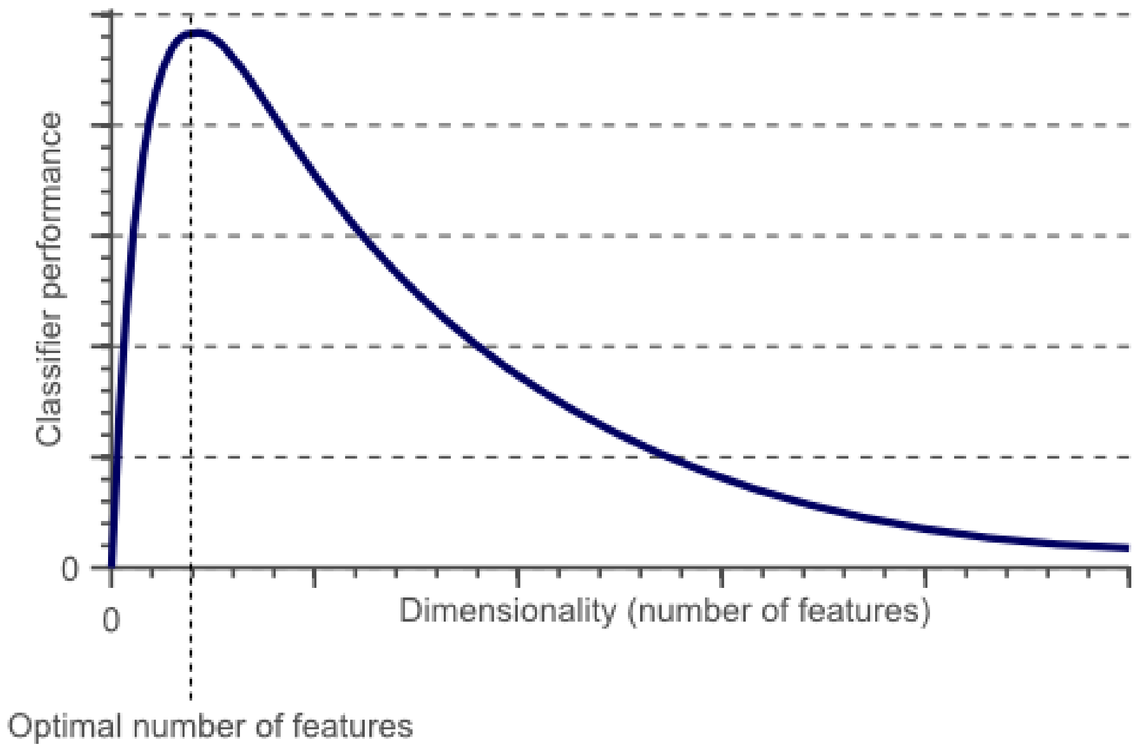
Bias-Variance trade-off





Final (important) words

Curse of dimensionality



QUESTIONS ?