LECTURE 5

Machine Learning & Decision Trees

Machine Learning

Entropy

Decision Trees (partially)

What is Learning?

Learning is any process by which a system improves its performance from experience

Herbert Simon

A computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P***, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**

Tom Mitchell

What to learn about language?

Assigning categories to words (part-of-speech [POS] tagging)
Assigning topics to articles, emails, or web pages
Mood, affect, or sentiment classification of a text or utterance
Assigning a semantic type or ontological class to a word or phrase
Language identification
Spoken word recognition
Handwriting recognition
Syntactic structure (sentence parsing)
Temporal ordering of historical events
Semantic roles for participants of events in a sentence
Named Entity (NE) identification
Coreference resolution
Discourse structure identification

Types of learning

Supervised learning

Unsupervised learning

Semi-supervised learning

Target function

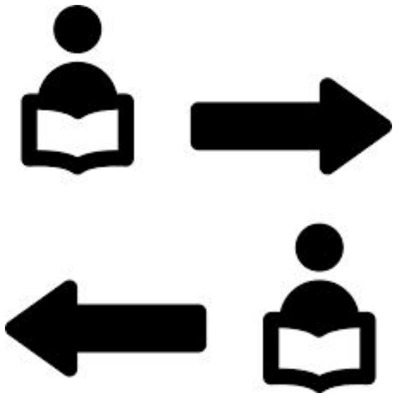
Target function maps input data to the desired output

Hypothesis (function) attempts to approximate the target function

Hypothesis space = a collection of all *possible* hypothesis functions

Learning from Experience = learning from training examples

Learning task



Learning involves improving on a task ***T*** with respect to a performance metric ***P***, based on experience ***E***

Tom Mitchell

Built corpus

Most informative and representative examples

Choose the training experience

Annotations increase available feature space

Identify the target function

How to represent the target function

Choose a learning algorithm

The way to infer the target function from the experience

Evaluate the results with the performance metric

Feature selection

Fix upon input for the target function

N-gram features

Structure-dependent features:

- Length; Nth element;

Annotation-dependent features – *new,* explicitly added information that can help in classification or discrimination.

- Person, organization, and Place

Target functions

Classification

- Binary (e.g. logistic regression)
  - span vs not-spam; sentiment analysis
- Multi-class (e.g. multinomial logistic regression)
  - natural language inference, genre detection

Probabilities of classes

Structure prediction

- Sequence labeling: POS tagging. segmentation
- Parsing: semantic & syntactic parsing

Regression analysis:

- Scalar value (i.e., a measure)
- Linear is the simplest

Price of properties
Degree of similarity
Essay grade

Types of learning (again)

Supervised learning

Unsupervised learning

Semi-supervised learning

Supervised learning

Data collection and annotation

Learning the target function

The most popular learning type

Unsupervised learning

Clustering

No annotated data

Identify naturally existing groupings in the dataset

Groups/clusters are not pre-defined (vs classification)

Contrast samples in the dataset to define clusters

Representation of the samples decides the nature of clusters

Types of clustering:

- Exclusive
- Overlapping: hierarchical

Semi-supervised (SS) learning

Combines pros of supervised & unsupervised methods:

- Supervised: annotated data is informative (but expensive)
- Unsupervised: ample availability of raw data but with less (explicit) info

Types of semi-supervised learning:

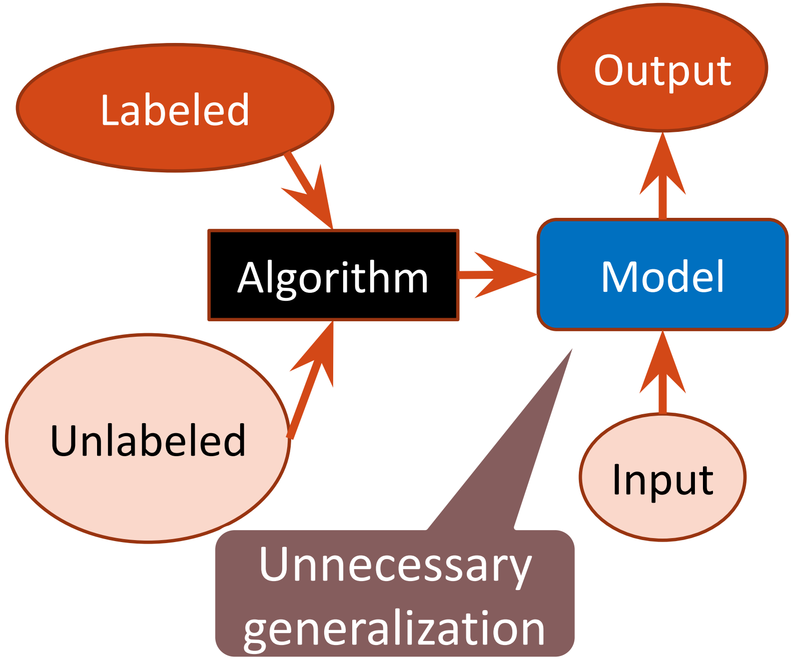
Active learning: human helps to label low-conf. samples

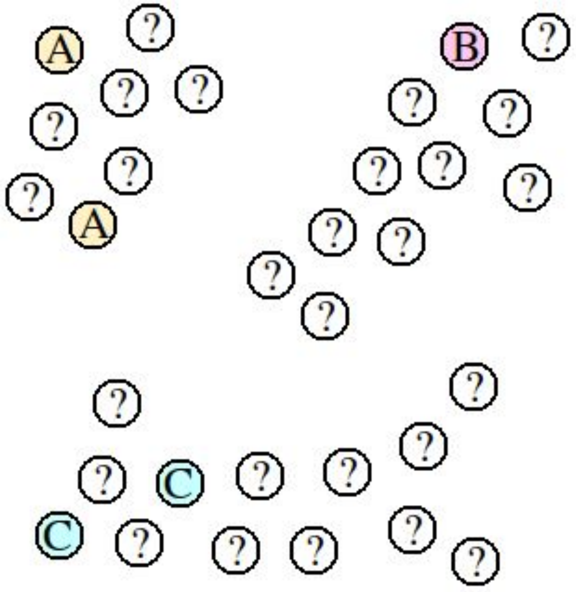
- Self-training: use for re-training unseen samples with high-conf. labels
- Multi-view: several ML models share with each other samples with high-conf. labels
- Self-ensemble: versions of an ML model voting or sharing samples with high-conf. labels

Inductive & transductive learning

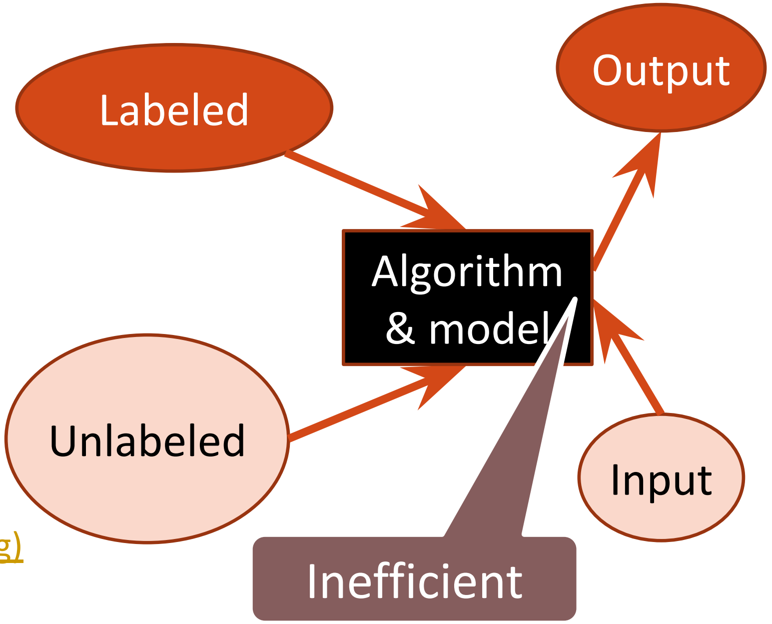
When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one.

Vladimir Vapnik





https://en.wikipedia.org/wiki/Transduction_(machine_learning)



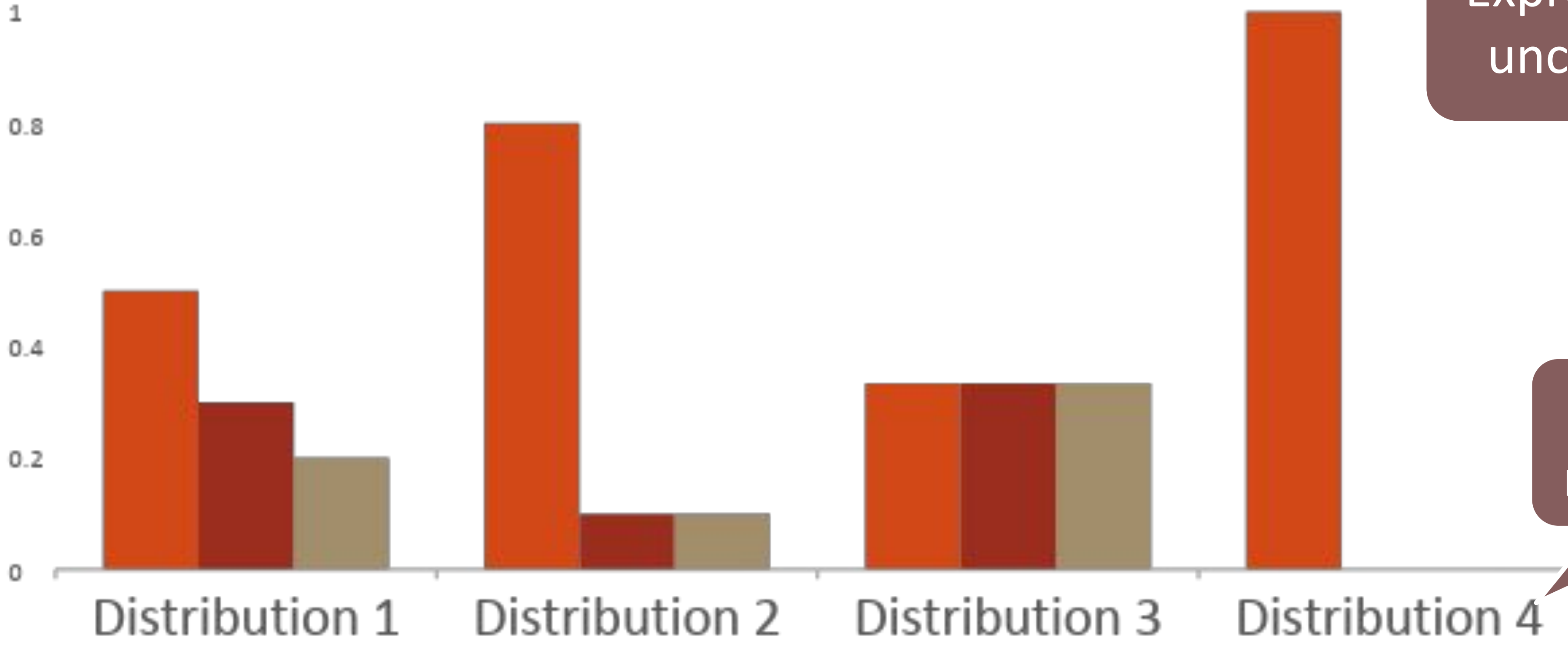
Understanding entropy

Entropy

The measure of uncertainty, chaos, mess, and diversity

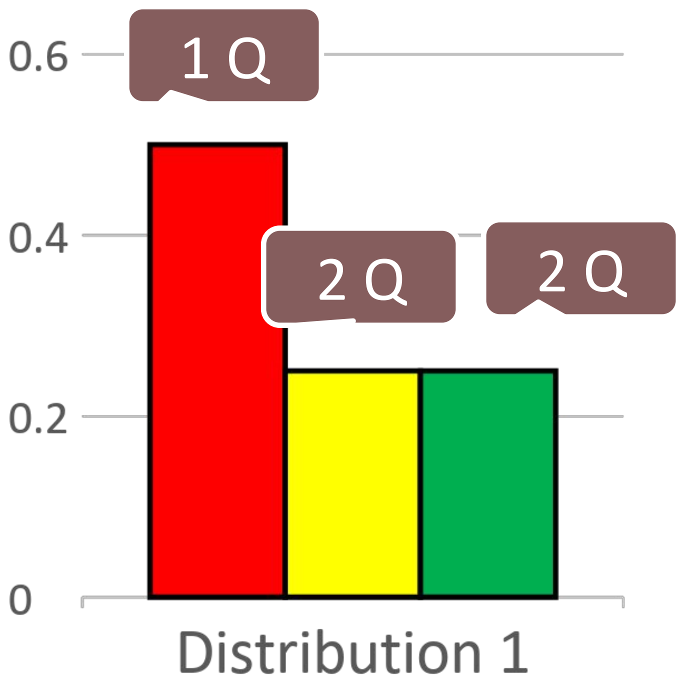
Expresses average uncertainty, etc.

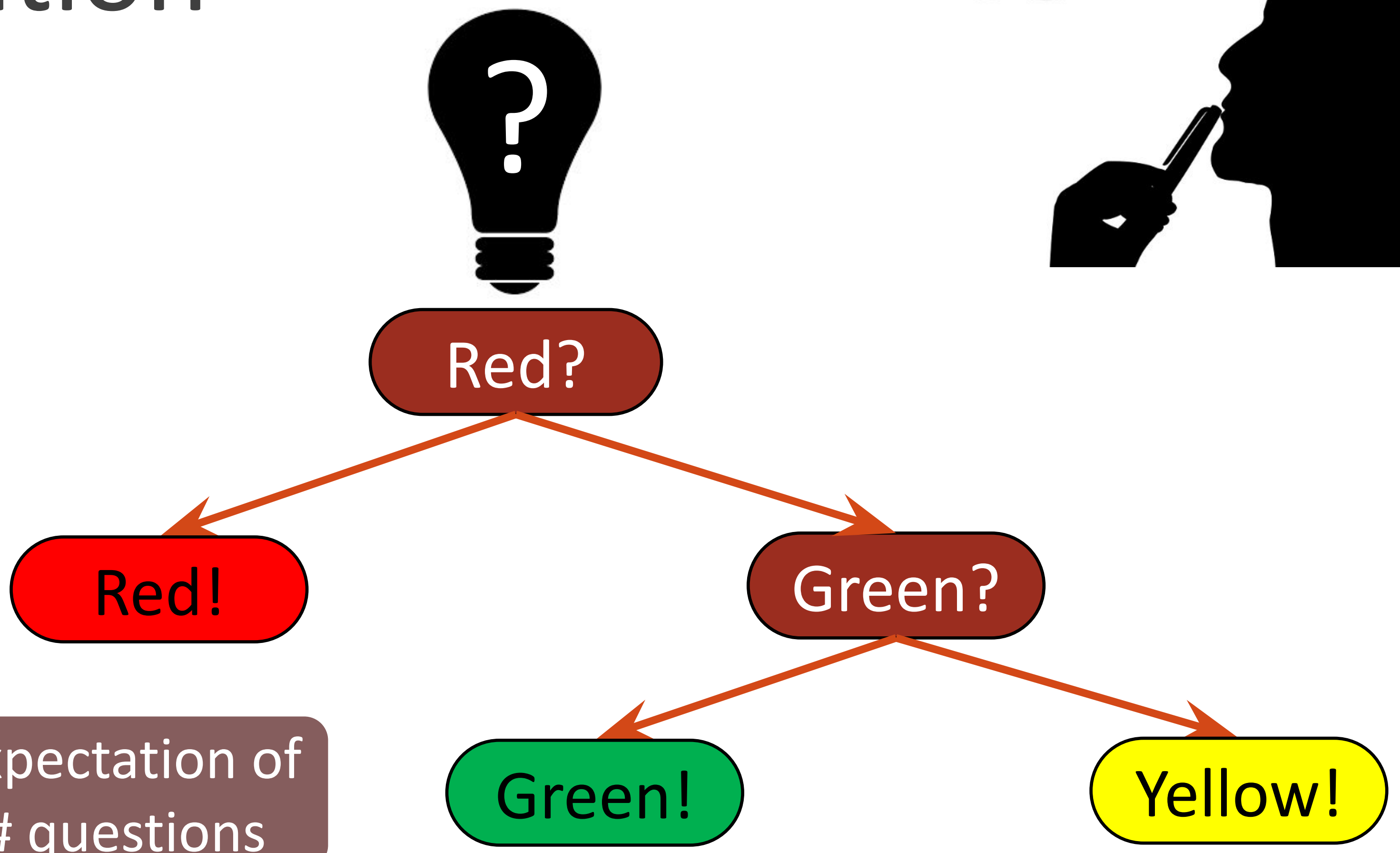
Which distributions are more uncertain?



Probability mass functions

Entropy: intuition





Expectation of # questions

$$\text{Avg. \# questions} = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2$$

Entropy: formula

The entropy of a discrete random variable X:

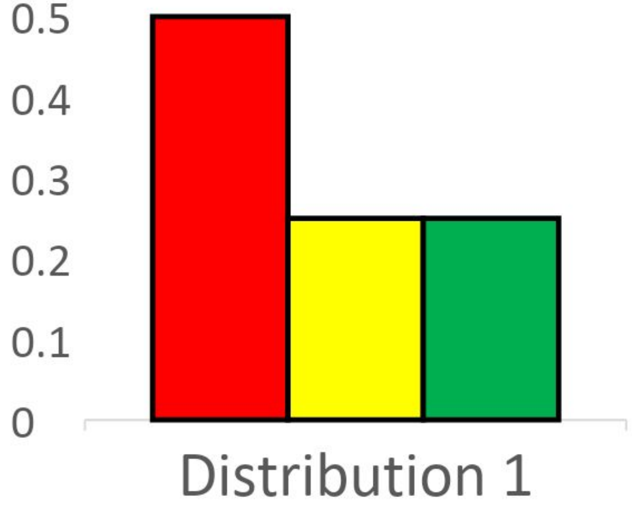
$$\boxed{H(X) = -\sum_{x \in V(X)} p(x) \log p(x) = \sum_{x \in V(X)} p(x)\, log \frac{1}{p(x)}}$$

Values of the random variable

Probability mass function

Base = 2
(serves as a scale)

Entropy doesn't depend on the values of the random variable



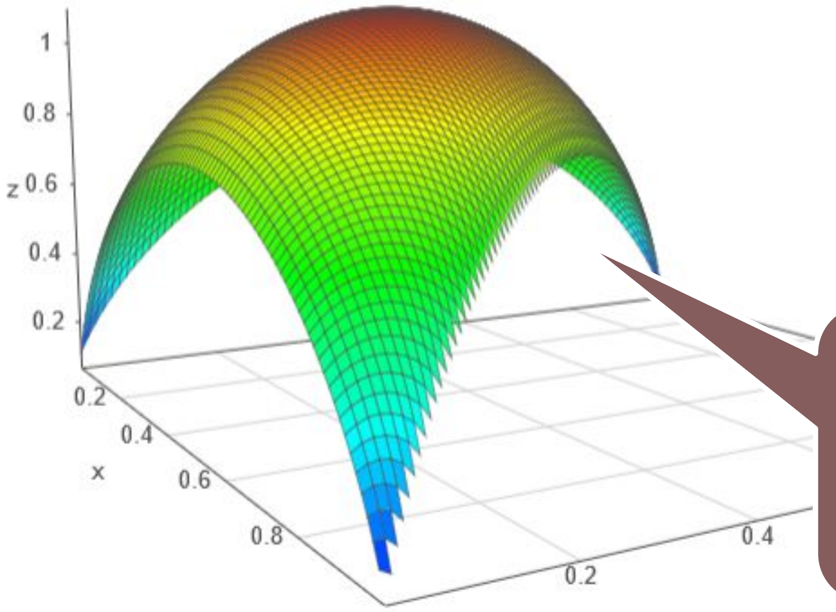
$$H(X)$$

$$\parallel$$

$$\text{Avg. \# questions} = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2$$

Entropy: two outcomes

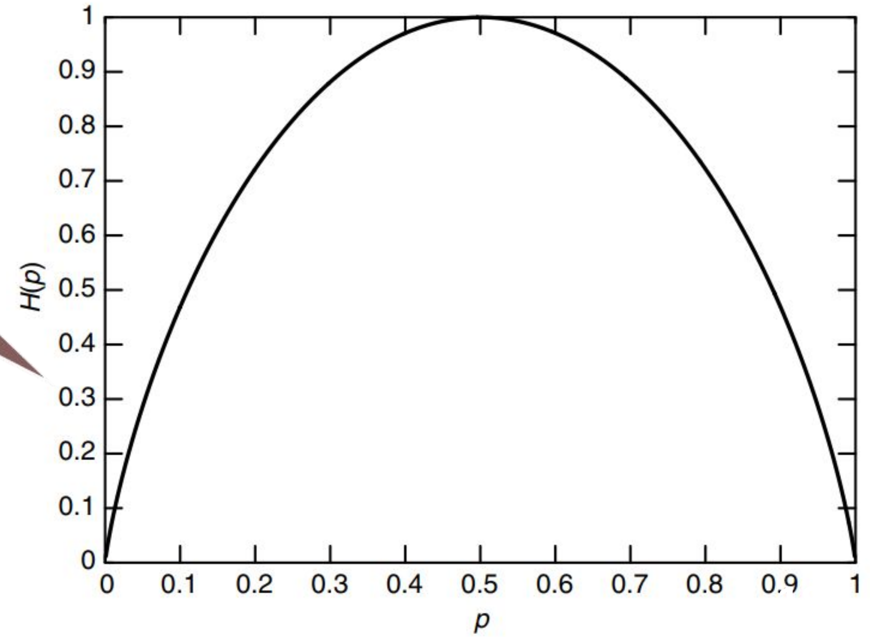
The entropy of a coin: $X = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p. \end{cases}$



Entropy for a three-valued random variable

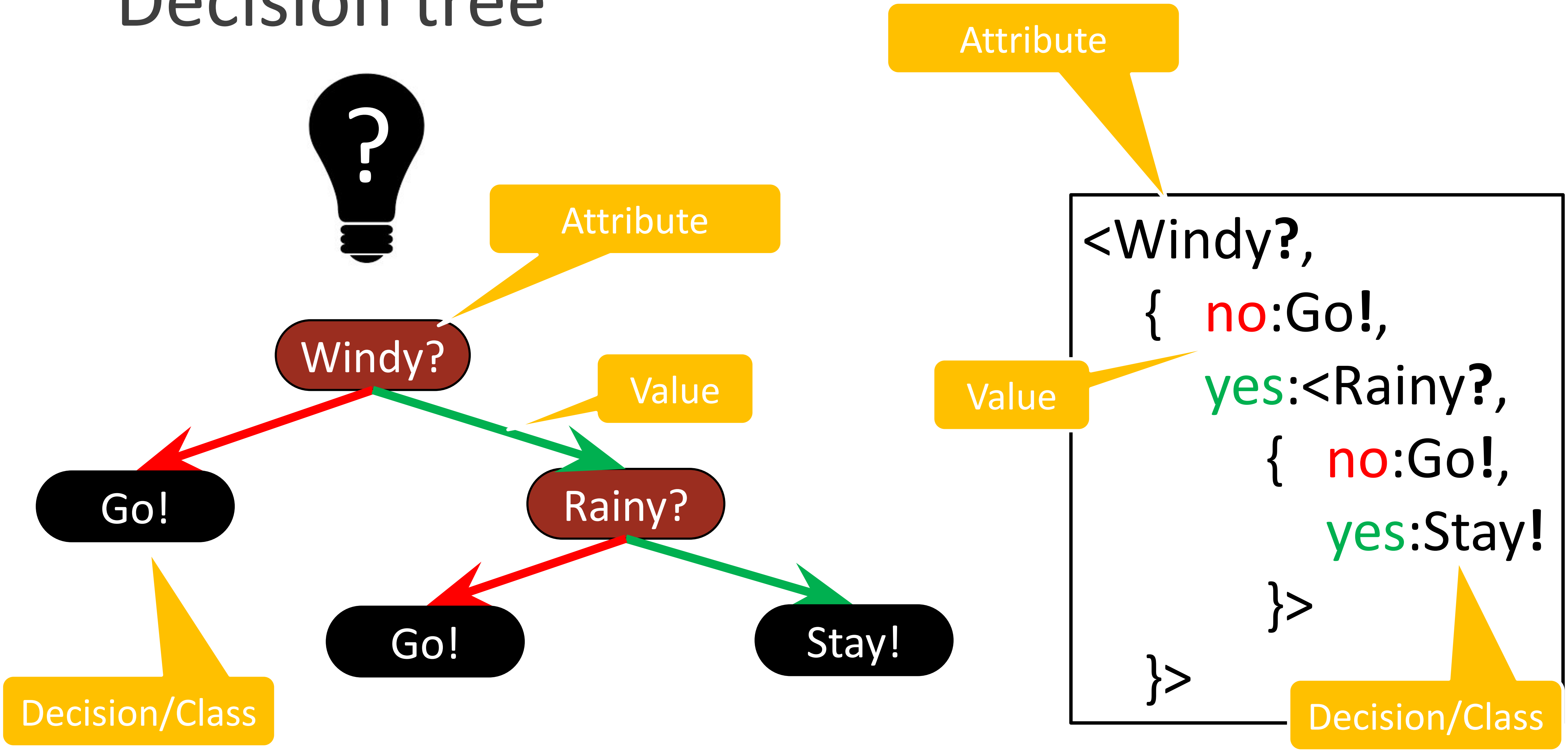
$0 \times \log 0 \stackrel{\text{def}}{=} 0$

Elements of Information theory (Cover & Thomas, 2006)



Decision Trees

Decision tree



ID3 algorithm (Quinlan, 1986)

```
ID3(Samples, Attributes)
    If all Samples are of some C class, return C!
    If Attributes = ∅, return most_common_class(Samples)!
    A := best_classifier_attribute(Attributes, Samples)
    R := <A?, ∅ >
    For a in values_of(A):
        If for no Samples, A=a:
                R[2].add(a: most_common_class(Samples)!)
        else:
                sub_Samples := Samples for which A=a
                less_Attributes := Attributes - A
                R[2].add(a: <ID3(sub_Samples, less_Attributes)> )
    return R
```

with classes

Best discriminating attribute for Samples from Attributes

Create a root of a decision tree

Recursive step: calling ID3 on less samples and less attributes

Information gain (with entropy)

Difference in avg. uncertainty level ≈ info

Info gained = avg. chaos before – avg. chaos now

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

Information gain

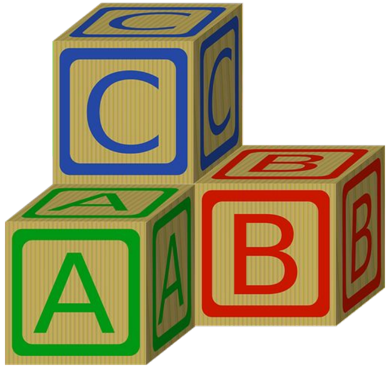
Entropy wrt the target class

Weight

best_classifier_attribute(Attributes, Samples) =

= $\mathrm{argmax}_{A \in Attributes}\, Gain(Samples, A)$

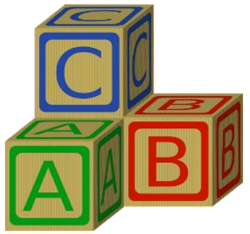
ID3 learning example



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

```
ID3(Samples, Attributes)
    If all Samples are of some C class, return C!
    If Attributes = ∅, return most_common_class(Samples)!
    A := best_classifier_attribute(Attributes, Samples)
    R := <A?, ∅ >
    For a in values_of(A):
        If for no Samples, A=a:
            R[2].add(a: most_common_class(Samples)!)
        else:
            sub_Samples := Samples for which A=a
            less_Attributes := Attributes - A
            R[2].add(a: <ID3(sub_Samples, less_Attributes)> )
    return R
```

ID3 learning example



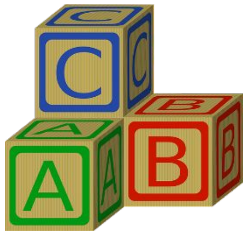
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

best_classifier_attribute(Attributes, Samples) =

$$= \text{argmax}_{A \in Attributes}\, Gain(Samples, A)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
https://www.saedsayad.com/decision_tree.htm				14

ID3 learning example



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Outlook

$$\textit{best\_classifier\_attribute}(\text{Attributes}, \text{Samples}) = \\ = \operatorname{argmax}_{A \in Attributes} Gain(Samples, A)$$

Gain = 0.247		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Gain = 0.029		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

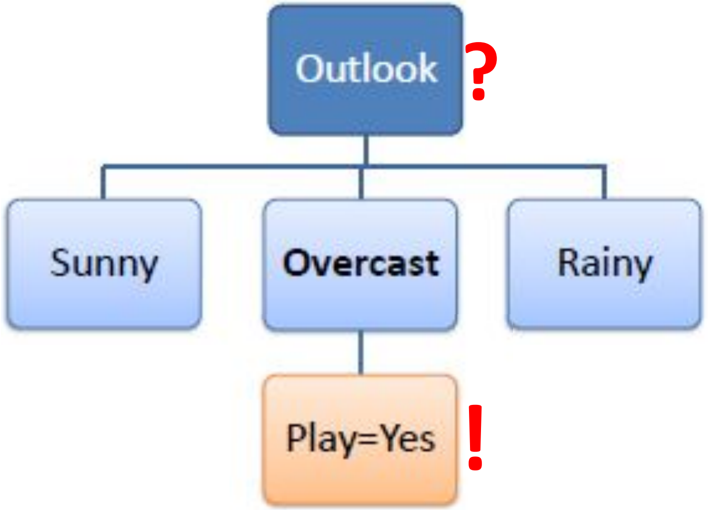
Gain = 0.152		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

Gain = 0.048		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

https://www.saedsayad.com/decision_tree.htm

ID3 learning example

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No




```
ID3(Samples, Attributes)
    If all Samples are of some C class, return C!
    If Attributes = Ø, return most_common_class(Samples)!
    A := best_classifier_attribute(Attributes, Samples)
    R := <A?, Ø >
    For a in values_of(A):
        If for no Samples, A=a:
            R[2].add(a: most_common_class(Samples)!)
        else:
            sub_Samples := Samples for which A=a
            less_Attributes := Attributes - A
            R[2].add(a: <ID3(sub_Samples, less_Attributes)> )
    return R
```

Outlook

Overcast

ID3 learning example

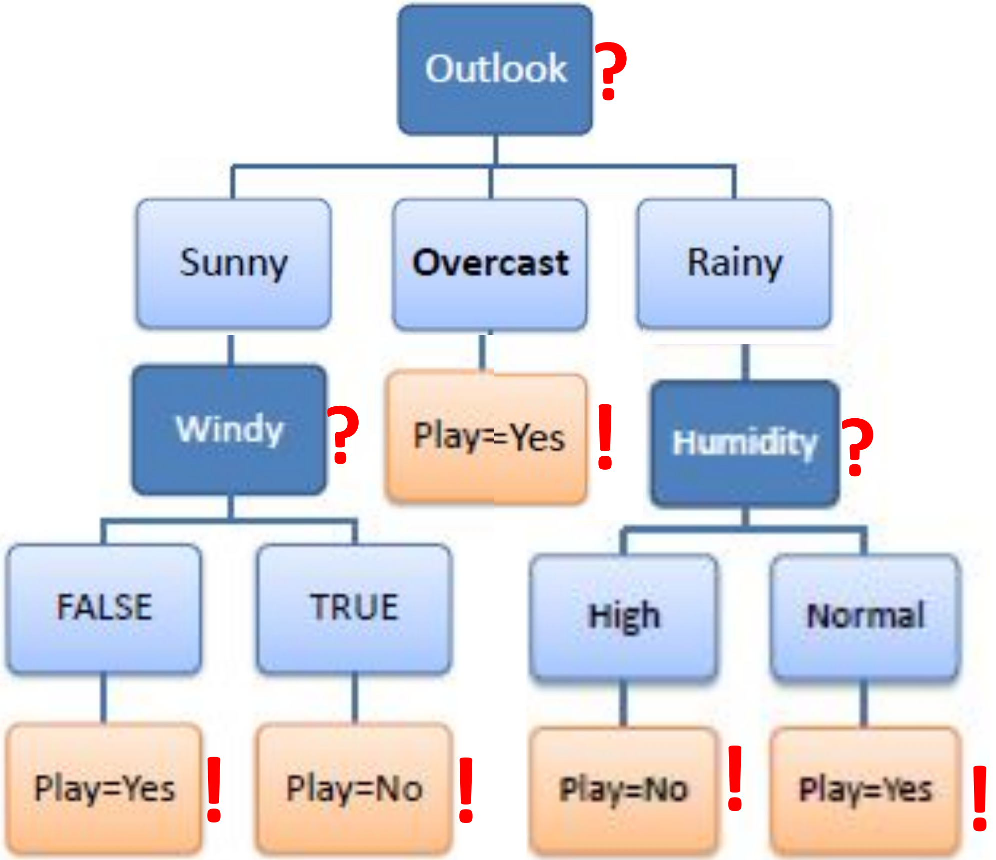
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No


```
ID3(Samples, Attributes)
    If all Samples are of some C class, return C!
    If Attributes = ∅, return most_common_class(Samples)!
    A := best_classifier_attribute(Attributes, Samples)
    R := <A?, ∅ >
    For α in values_of(A):
        If for no Samples, A=α:
            R[2].add(α: most_common_class(Samples)!)
        else:
            sub_Samples := Samples for which A=α
            less_Attributes := Attributes - A
            R[2].add(α: <ID3(sub_Samples, less_Attributes)> )
    return R
```

Windy

ID3 learning example

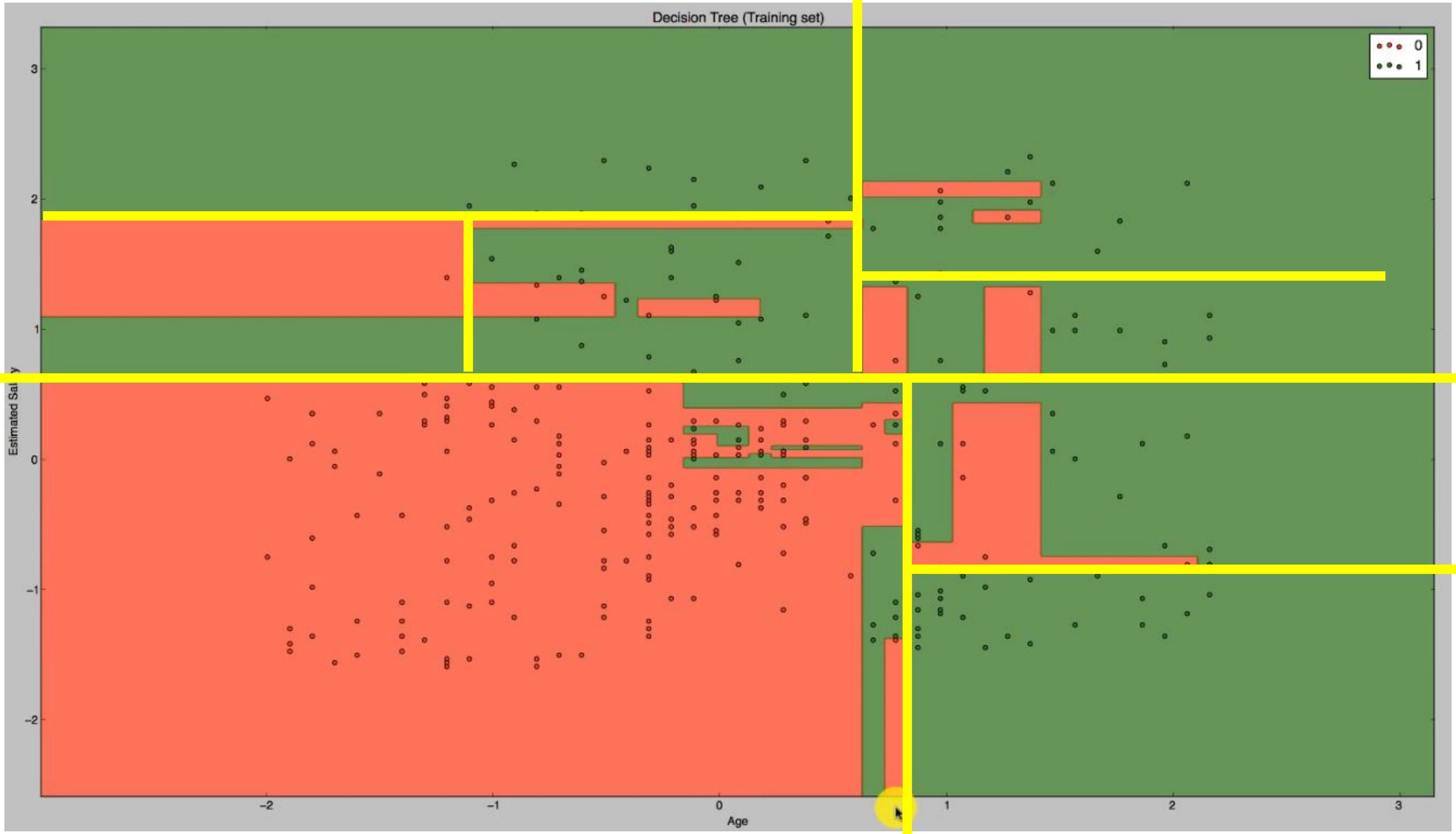
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

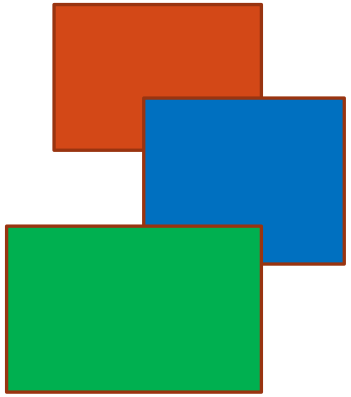


Humidity

```
ID3(Samples, Attributes)
    If all Samples are of some C class, return C!
    If Attributes = ∅, return most_common_class(Samples)!
    A := best_classifier_attribute(Attributes, Samples)
    R := <A?, ∅ >
    For α in values_of(A):
        If for no Samples, A=α:
            R[2].add(α: most_common_class(Samples)!)
        else:
            sub_Samples := Samples for which A=α
            less_Attributes := Attributes - A
            R[2].add(α: <ID3(sub_Samples, less_Attributes)> )
    return R
```

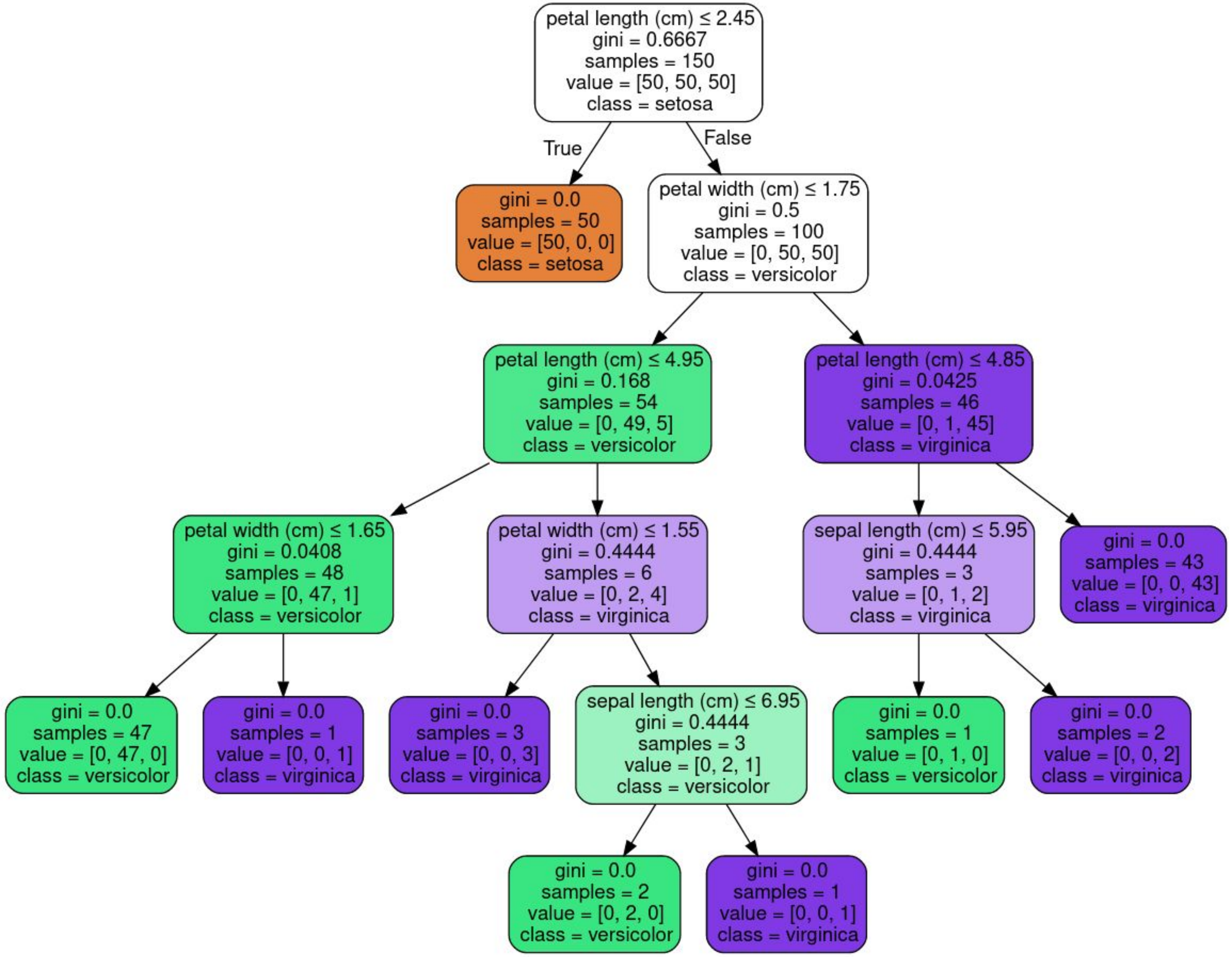
What decision trees actually do





Machine Learning A-Z™: Hands-On Python & R In Data Science https://www.udemy.com/machinelearning/

Further Reading



https://scikit-learn.org/stable/modules/tree.html