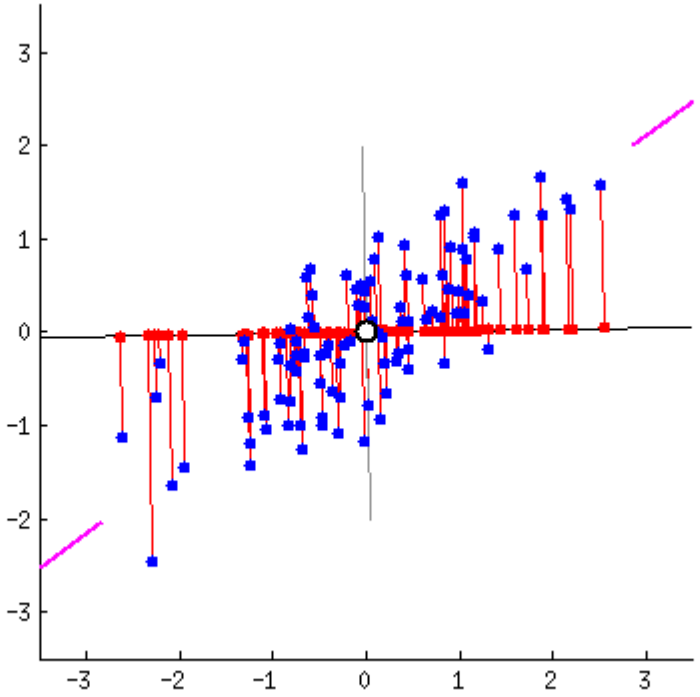
Linear Discriminant Analysis

- Dimensionality reduction technique often used for supervised classification
- PCA good for feature vectors of high dimension with new features of lower dimension by taking linear combinations of original features
- But, in general the direction in which there is much variability is not necessarily useful for "discriminating" between samples belonging to different classes (or labels)
- Suppose that we want to "project" data $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$ (feature vectors) in *d* dimensional feature space onto a line and each feature vector belongs to one of two classes $\mathcal{D}_1$ or $\mathcal{D}_2$
- Question: Which line should we choose?

Linear Discriminant Analysis

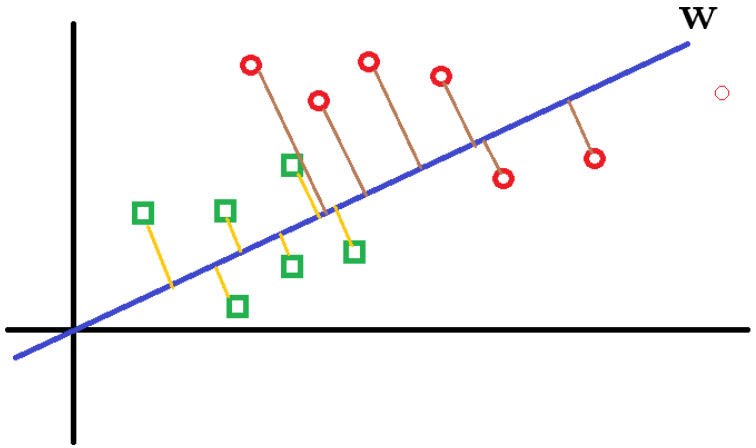
- A picture is worth a thousand words … maybe more …



- Choosing a line is equivalent selecting a unit vector along the line

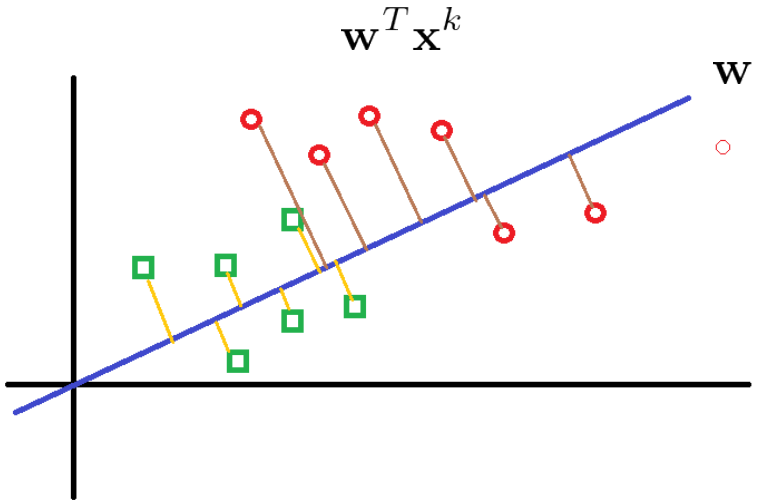
Linear Discriminant Analysis

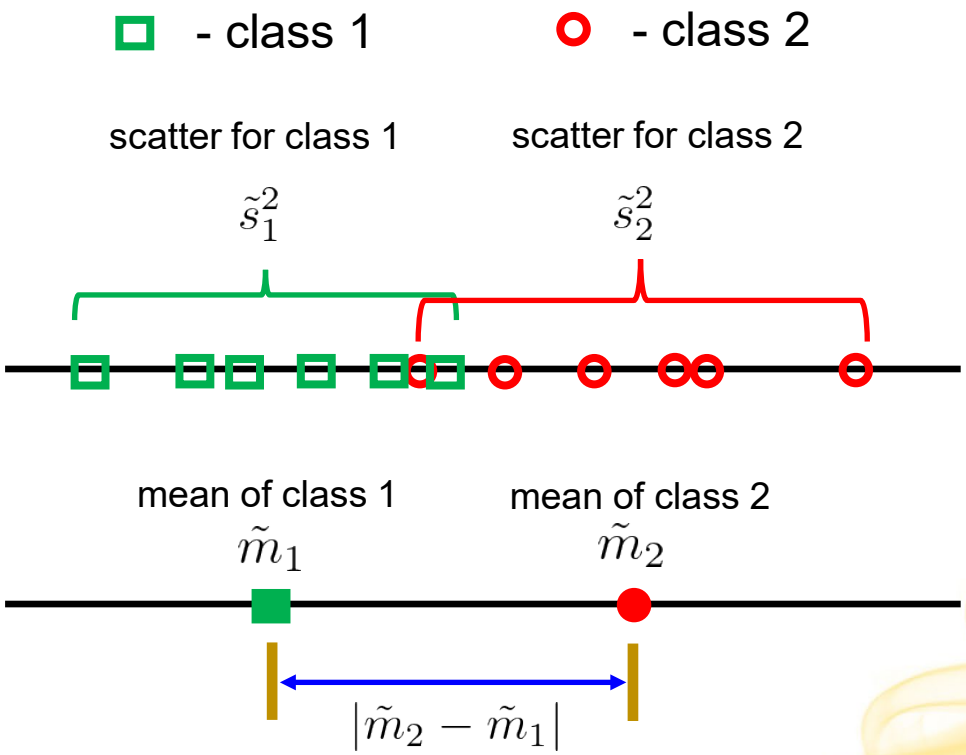
- Assumptions
  - Data/samples have (jointly) Gaussian distributions
  - Data/samples linearly separable, i.e., a straight line or a decision boundary can be drawn to separate data/samples
  - Classes have identical covariance matrix
- Criteria for LDA
  - Maximize the distance between the means of two classes
  - Minimize the variance within individual classes



Linear Discriminant Analysis

- Example





$$\text{maximize } J(\mathbf{w}) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

within-class scatter

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y, \qquad \tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2$$

Linear Discriminant Analysis

- Question: How do we choose the line or, equivalently, the vector $\mathbf{w}$, which maximizes

$$J(\mathbf{w}) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- Answer: $\mathbf{w} = \mathbf{S}_{\mathbf{w}}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$, where

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{x \in \mathcal{D}_i} \mathbf{x}, i = 1, 2 \quad \text{with } n_i = |\mathcal{D}_i|$$

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2, \quad \text{and} \quad \mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

within-class scatter matrix

Linear Discriminant Analysis

- Boring algebra ….

$$\tilde{s}_i^2 = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_i)^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2 = \mathbf{w}^T \mathbf{S}_i \mathbf{w} \quad \Rightarrow \quad \tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T \mathbf{S}_W \mathbf{w}$$

$$\begin{aligned}(\tilde{m}_1 - \tilde{m}_2)^2 &= (\mathbf{w}^T \mathbf{m}_1 - \mathbf{w}^T \mathbf{m}_2)^2 = \mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w} \\ &= \mathbf{w}^T \mathbf{S}_B \mathbf{w}, \quad \text{where} \quad \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\end{aligned}$$

between-class scatter matrix

$$\Rightarrow \quad J(\mathbf{w}) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- Generalized Rayleigh quotient

Maximizer satisfies $\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$ for some constant λ

Linear Discriminant Analysis

- When $\mathbf{S}_W$ is invertible,

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w} \quad \text{for some constant } \lambda \quad \Longleftrightarrow \quad \mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w} \quad \text{for some constant } \lambda$$

Always lies in the direction of $\mathbf{m}_1 - \mathbf{m}_2$ because $\mathbf{S}_B\mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w}$

$$\beta\mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) = \lambda\mathbf{w}$$

(not worrying about scalar) $\Rightarrow$ $\mathbf{w} = \mathbf{S}_{\mathbf{w}}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$

NEAREST NEIGHBOR RULES DECISION TREE

DATA/MSML 603: Principles of Machine Learning

Supervised Learning: Classification

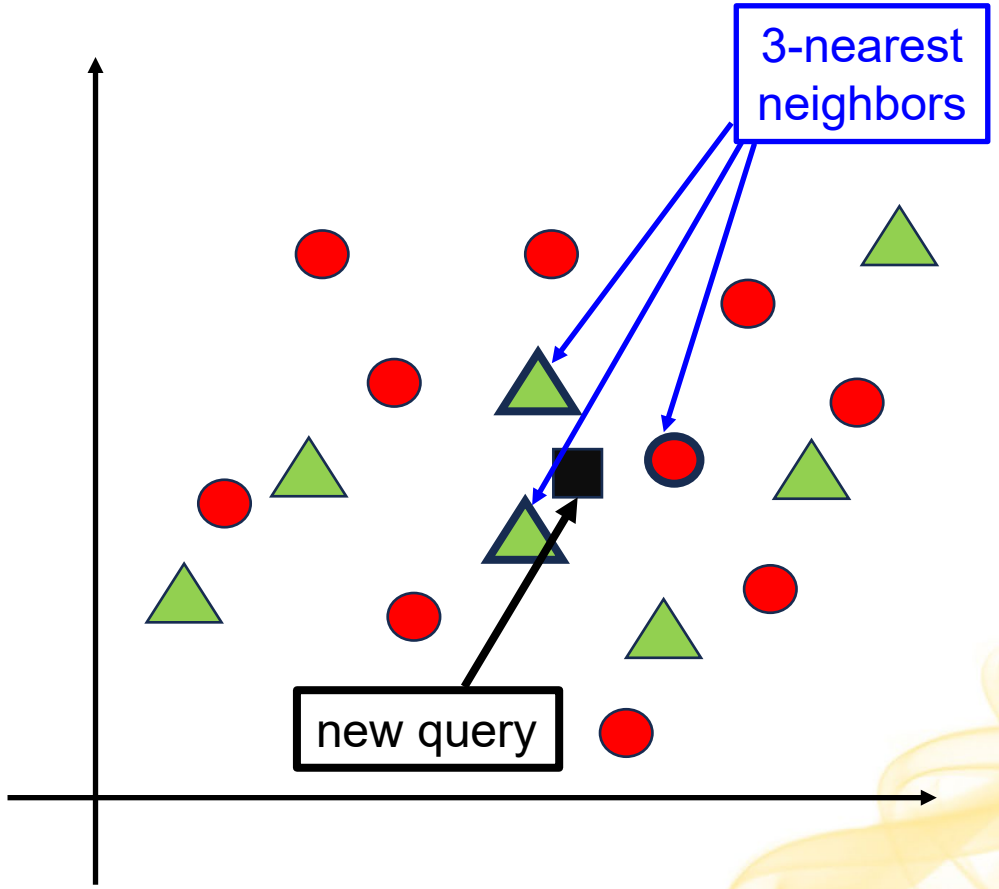
- k-nearest neighbors (kNN) algorithm
- Decision tree
- Perceptron
- Support vector machine

k-Nearest Neighbors Algorithm

- Example of instance-based learning method
  - Simply stores the training samples (called “lazy” learning model)
  - Generalization performed only when a new instance must be classified (no explicit learning is performed to construct a target function used to predict results)
  - Each time a new query instance is generated, its relationship to the previously stored samples is used to assign a target function value for the new instance

k-Nearest Neighbors Algorithm

- Given a new query (or instance),
  - Find *k* nearest neighbors (according to a chosen distance metric)
  - Choose the label most common among the *k* nearest neighbors (plurality vote or majority vote)



k-Nearest Neighbors Algorithm

- Distance metrics
  - Euclidean distance – most commonly used

$$d_E(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{k=1}^{d} (x_k^1 - x_k^2)^2} = \|\mathbf{x}^1 - \mathbf{x}^2\|_2$$

  - Manhattan distance (also known as taxicab distance)

$$d_M(\mathbf{x}^1, \mathbf{x}^2) = \sum_{k=1}^{d} \left|x_k^1 - x_k^2\right| = \|\mathbf{x}^1 - \mathbf{x}^2\|_1$$

k-Nearest Neighbors Algorithm

- Distance metrics (cont'd)
  - Minkowski distance – generalized of Euclidean and Manhattan distances

$$d_{MI}(\mathbf{x}^1, \mathbf{x}^2) = \left(\sum_{k=1}^{d} |x_k^1 - x_k^2|^p \right)^{1/p} = \|\mathbf{x}^1 - \mathbf{x}^2\|_p$$

  - Hamming distance

$$d_H(\mathbf{x}^1, \mathbf{x}^2) = \sum_{k=1}^{d} \mathbf{1}\{x_k^1 \neq x_k^2\}$$

Distance-weighted Nearest Neighbors Algorithm

- Refinement to k-nearest neighbor algorithm
  - Contribution of each of the k nearest neighbors weighed according to their distance to the query point

k-Nearest Neighbors Algorithm

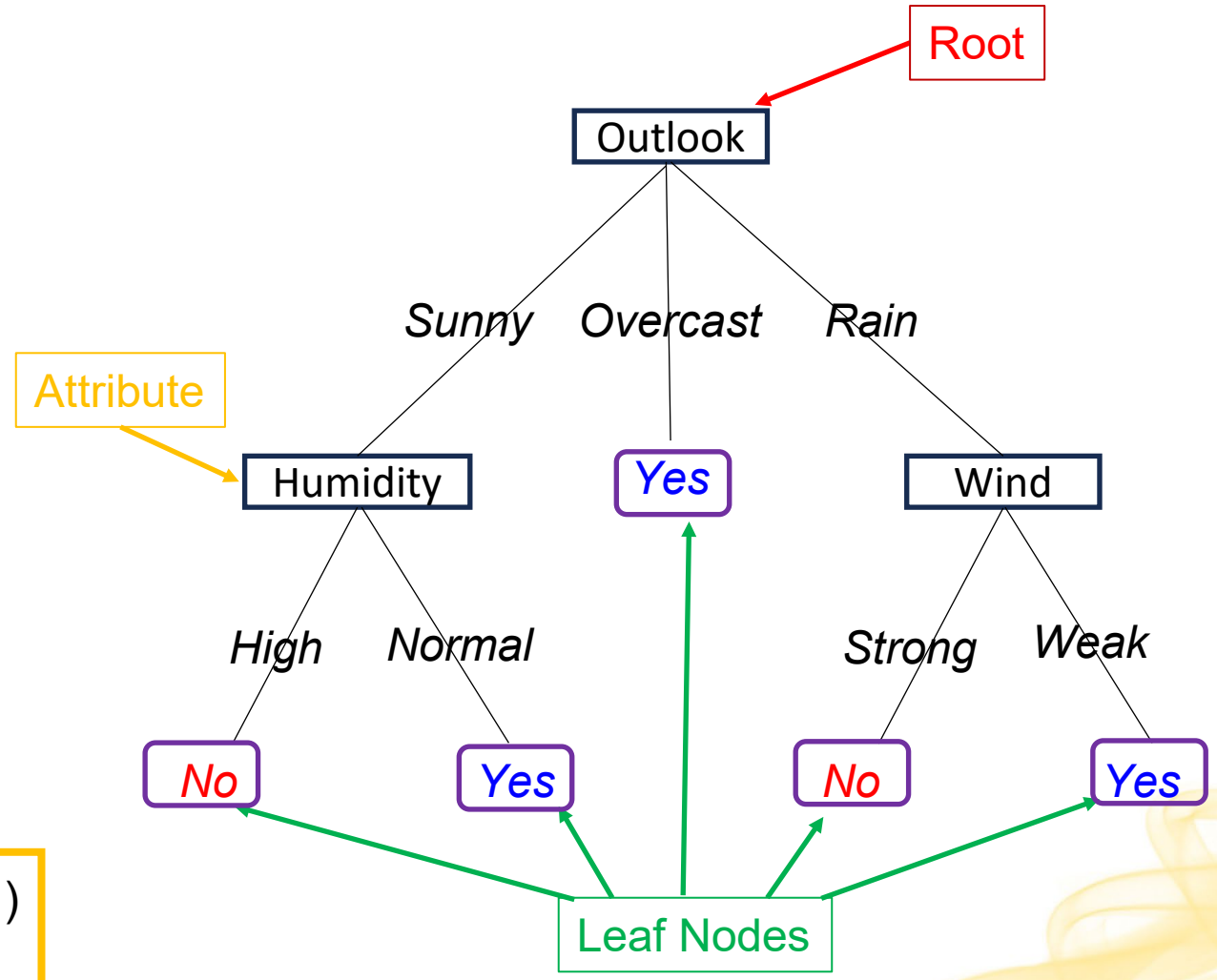
- Strengths
  - Robust to noisy training data – Impact of isolated noisy training samples can be smoothed out
  - Effective when a large set of training data is available

Decision Trees

- One of the most widely used and practical methods for inductive inference
- Used to approximate discrete-valued target functions
- Advantages
  - Robust to noisy data
  - Easy to interpret
  - Flexible
- Can be represented either as a (decision) tree or a set of if-then rules

Decision Trees

- Classifies instances by sorting them down the tree from the root to some leaf node
- Example: Classifies Saturday mornings according to whether or not they are suitable for playing tennis



(Outlook = Sunny $\wedge$ Humidity = Normal)

$\vee$ (Outlook = Overcast)

$\vee$ (Outlook = Rain $\wedge$ Wind = Weak)

< Outlook = Sunny, Temperature = Hot, Humidity = High, Wind= Strong >

Decision Trees: Basic Idea

- Decision tree constructed by asking a sequence of questions using attributes, starting with the root (node)
  - Each question leads to splitting, leading to a branch/subtree or a leaf node
- Question: Which question should we ask first?
  - Intuitively, we should choose the question that does the best job of classifying the training examples by itself
  - In other words, choose the attribute/question that is the “best classifier”
- Two popular measures of quality of the split in the tree
  - Gini impurity
  - Information gain

Gini Impurity

- Suppose that there are c classes and the fraction of class j is $p_j,\ j = 1, \ldots, c$

$$\text{Gini Impurity} = \sum_{j=1}^{c} p_j(1 - p_j) = 1 - \sum_{j=1}^{c} p_j^2$$

  - Measures the frequency at which any element of the dataset will be mislabeled when it is randomly labeled

Entropy and Information Gain

- Suppose that there are c classes and the fraction of class j is $p_j,\ j = 1, \ldots, c$

$$\text{Entropy} = -\sum_{j=1}^{c} p_j \ \log_2(p_j)$$

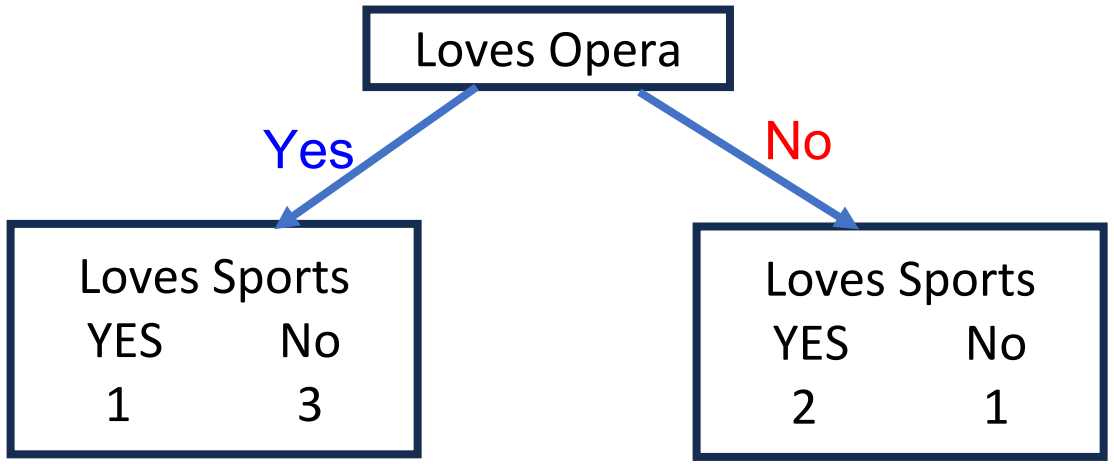
  - Measures the amount of uncertainty or information associated with potential outcome

- Information gain = Entropy before splitting – weight average entropy after splitting
  - Expected reduction in entropy induced by partitioning the training examples according to the attribute or question

$$\text{Information Gain}(S) = \text{Entropy}(S) - \sum_{a \in Answers} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Decision Trees: Example

- Gini Impurity for “Loves Opera”



$$\text{GI} = 1 - (1/4)^2 - (3/4)^2 = 0.375$$

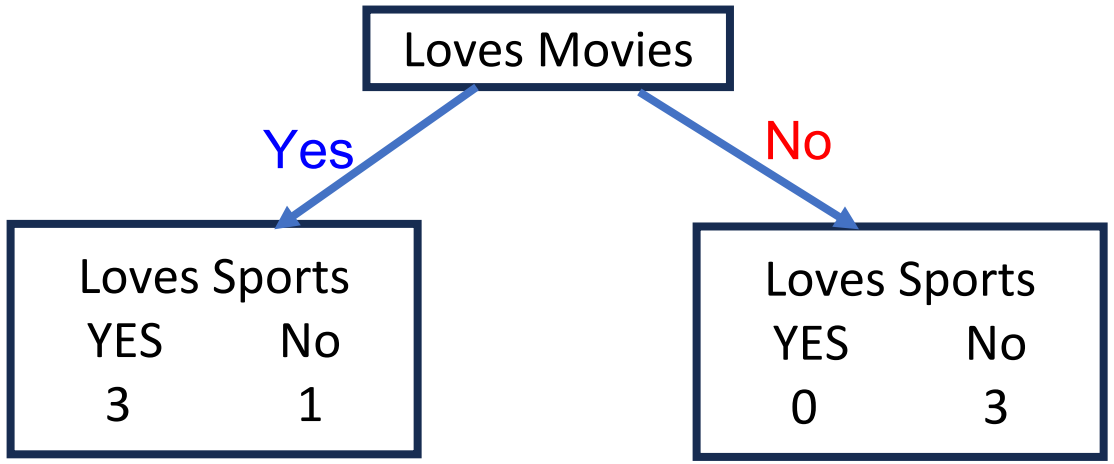
$$\text{GI} = 1 - (2/3)^2 - (1/3)^2 = 0.444$$

$$\text{Total GI} = \frac{4}{7}0.375 + \frac{3}{7}0.444 = 0.405$$

Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

Decision Trees: Example

- Gini Impurity for “Loves Movies”



$$\text{GI} = 1 - (3/4)^2 - (1/4)^2 = 0.375$$

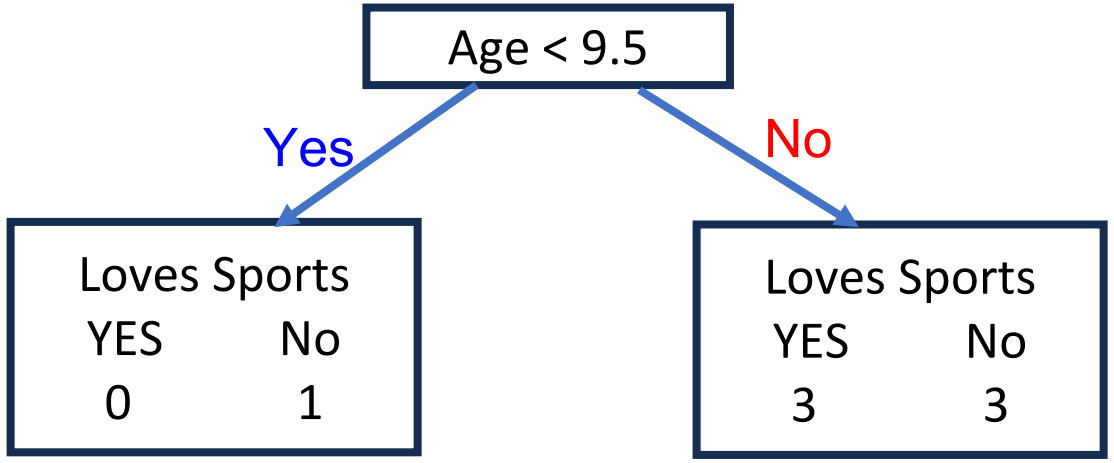
$$\text{GI} = 1 - (0/3)^2 - (3/3)^2 = 0$$

$$\text{Total GI} = \frac{4}{7}0.375 + \frac{3}{7}0 = 0.214$$

Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

Decision Trees: Example

- Gini Impurity for “Age”



	Age	Loves Sports
9.5	7	No
15	12	No
26.5	18	Yes
36.5	35	Yes
44	38	Yes
66.5	50	No
	83	No

$$\text{GI} = 1 - (0/1)^2 - (1/1)^2 = 0$$

$$\text{GI} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$\text{Total GI} = \frac{1}{7}0 + \frac{6}{7}0.5 = 0.429$$

Decision Trees: Example

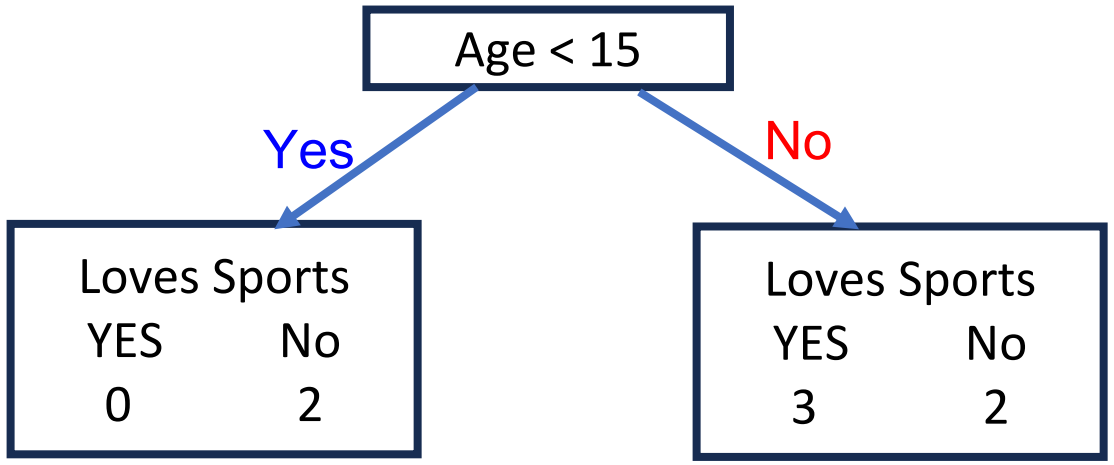
- Gini Impurity for “Age”

Age < 9.5	Gini Impurity = 0.429
Age < 15	Gini Impurity = 0.343
Age < 26.5	Gini Impurity = 0.476
Age < 36.5	Gini Impurity = 0.476
Age < 44	Gini Impurity = 0.343
Age < 66.5	Gini Impurity = 0.429

	Age	Loves Sports
9.5	7	No
15	12	No
26.5	18	Yes
36.5	35	Yes
44	38	Yes
66.5	50	No
	83	No

Decision Trees: Example

- Gini Impurity for “Age”



$$\text{GI} = 1 - (0/2)^2 - (2/2)^2 = 0$$

$$\text{GI} = 1 - (3/5)^2 - (2/5)^2 = 0.480$$

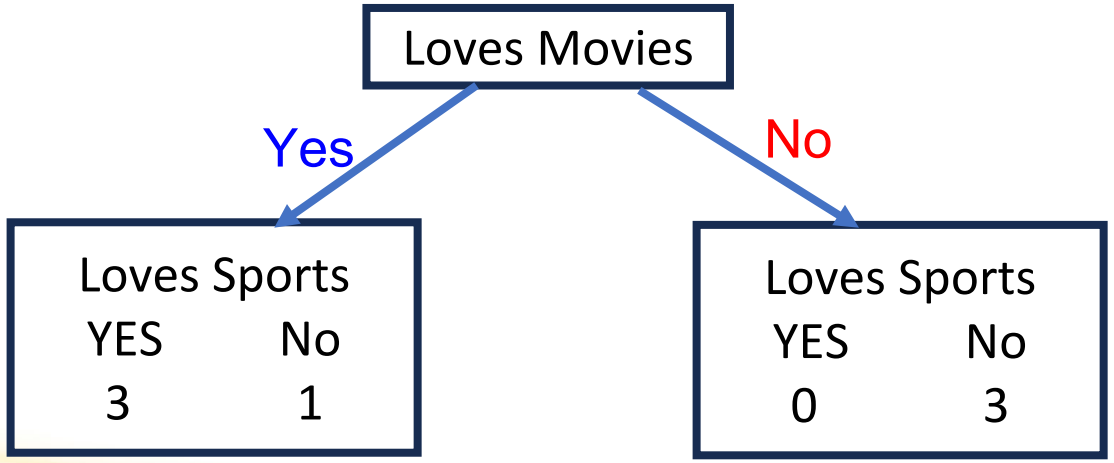
$$\text{Total GI} = \frac{2}{7}0 + \frac{5}{7}0.480 = 0.343$$

9.5
15
26.5
36.5
44
66.5

Age	Loves Sports
7	No
12	No
18	Yes
35	Yes
38	Yes
50	No
83	No

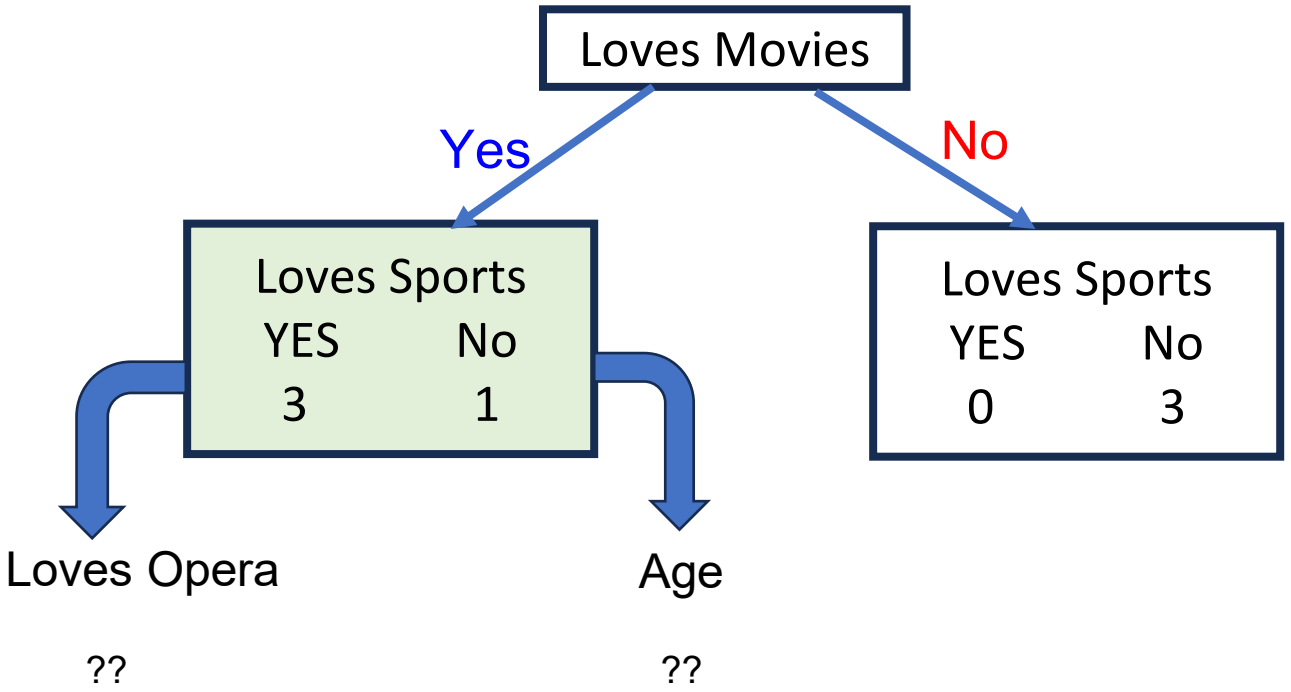
Decision Trees: Example

- Gini Impurity for “Loves Opera” = 0.405
- Gini Impurity for “Loves Movies” = 0.214
- Gini Impurity for “Age” = 0.343



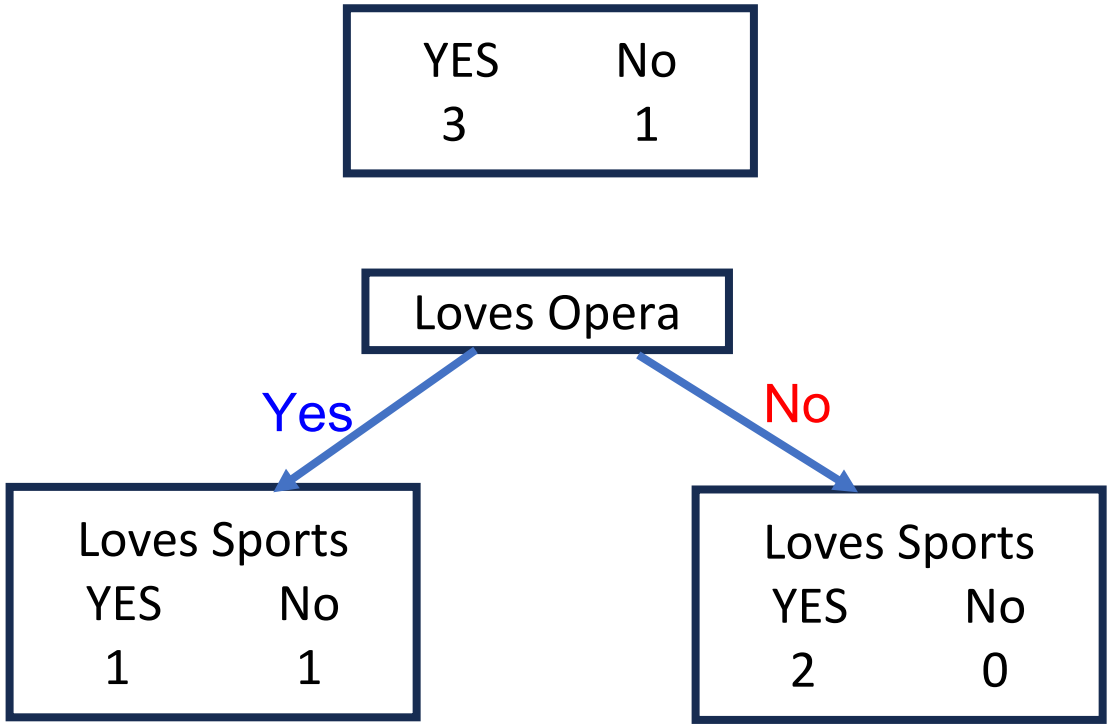
Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

Decision Trees: Example



Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

Decision Trees: Example



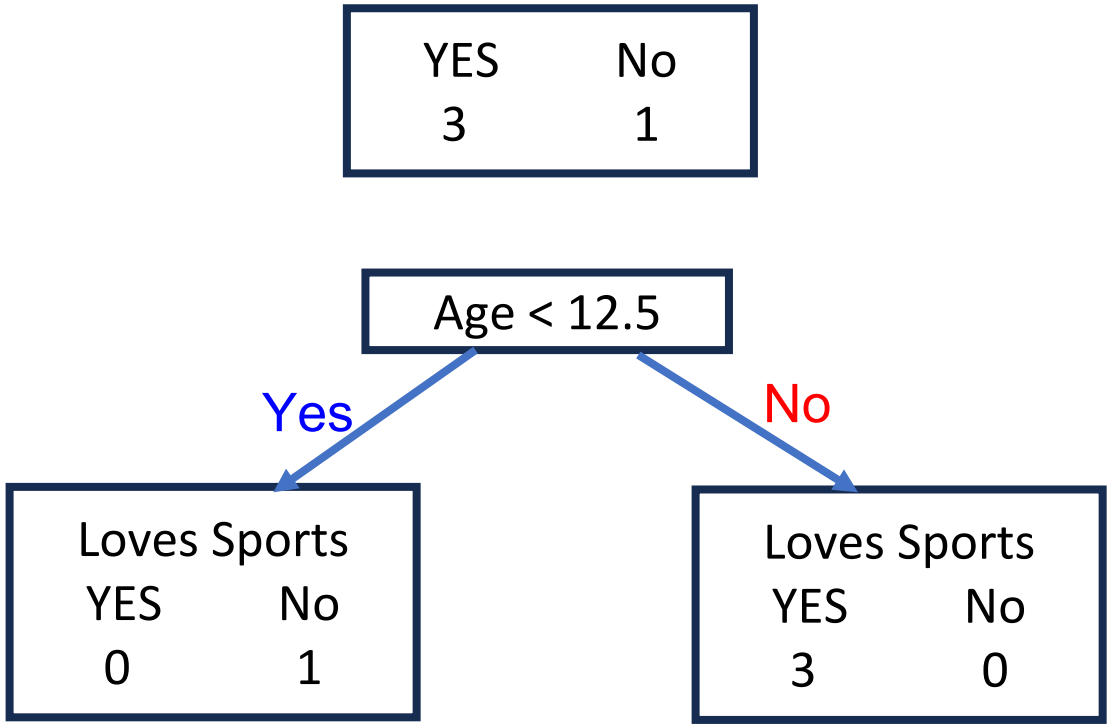
Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

$$\text{GI} = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$\text{GI} = 1 - (2/2)^2 - (0/2)^2 = 0$$

$$\text{Total GI} = \frac{2}{4}0.5 + \frac{2}{4}0 = 0.25$$

Decision Trees: Example



$$\text{Total GI} = 0$$

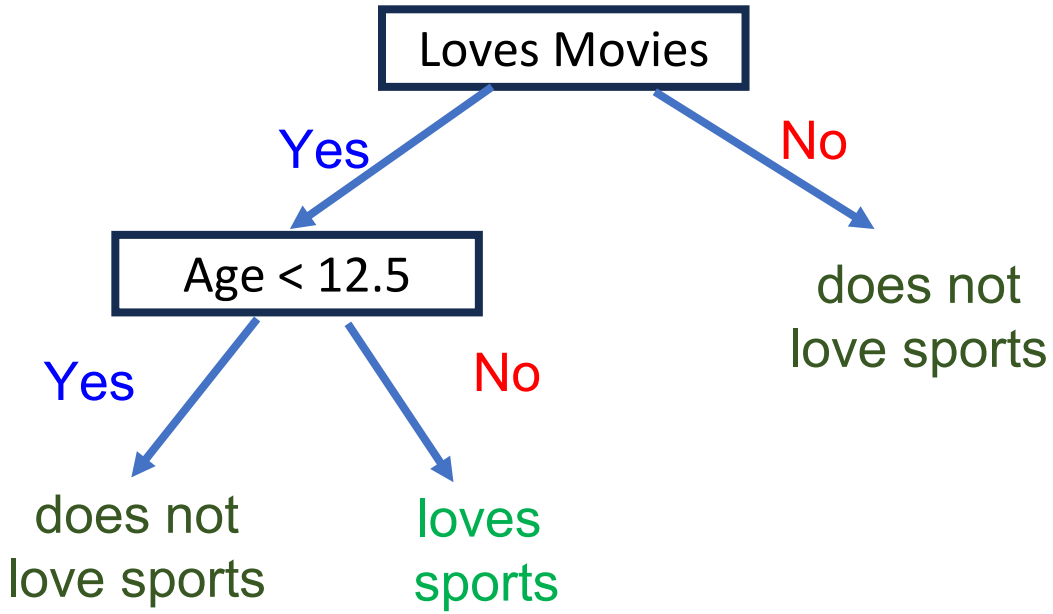
Age	Loves Sports
7	No
12	No
18	Yes
35	Yes
38	Yes
50	No
83	No

12.5

26.5

36.5

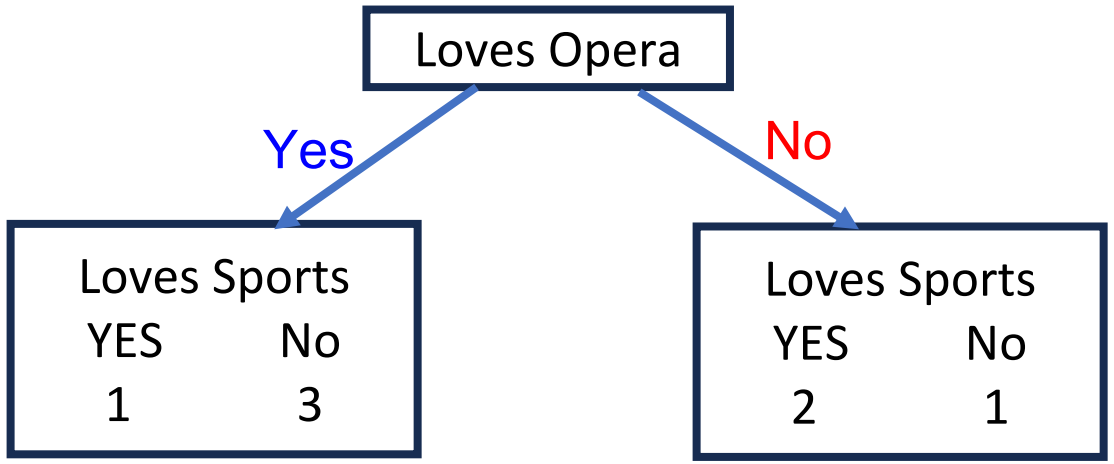
Decision Trees: Example



Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

Decision Trees: Example

- Gini Impurity for “Loves Opera”



Loves Opera	Loves Movies	Age	Loves Sports
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

$$\text{Entropy} = -((1/4)\log_2(1/4) + (3/4)\log_2(3/4)) = 0.811$$

$$\text{Entropy} = -((2/3)\log_2(2/3) + (1/3)\log_2(1/3)) = 0.918$$

$$\text{Entropy} = -((3/7)\log_2(3/7) + (4/7)\log_2(4/7)) = 0.985$$

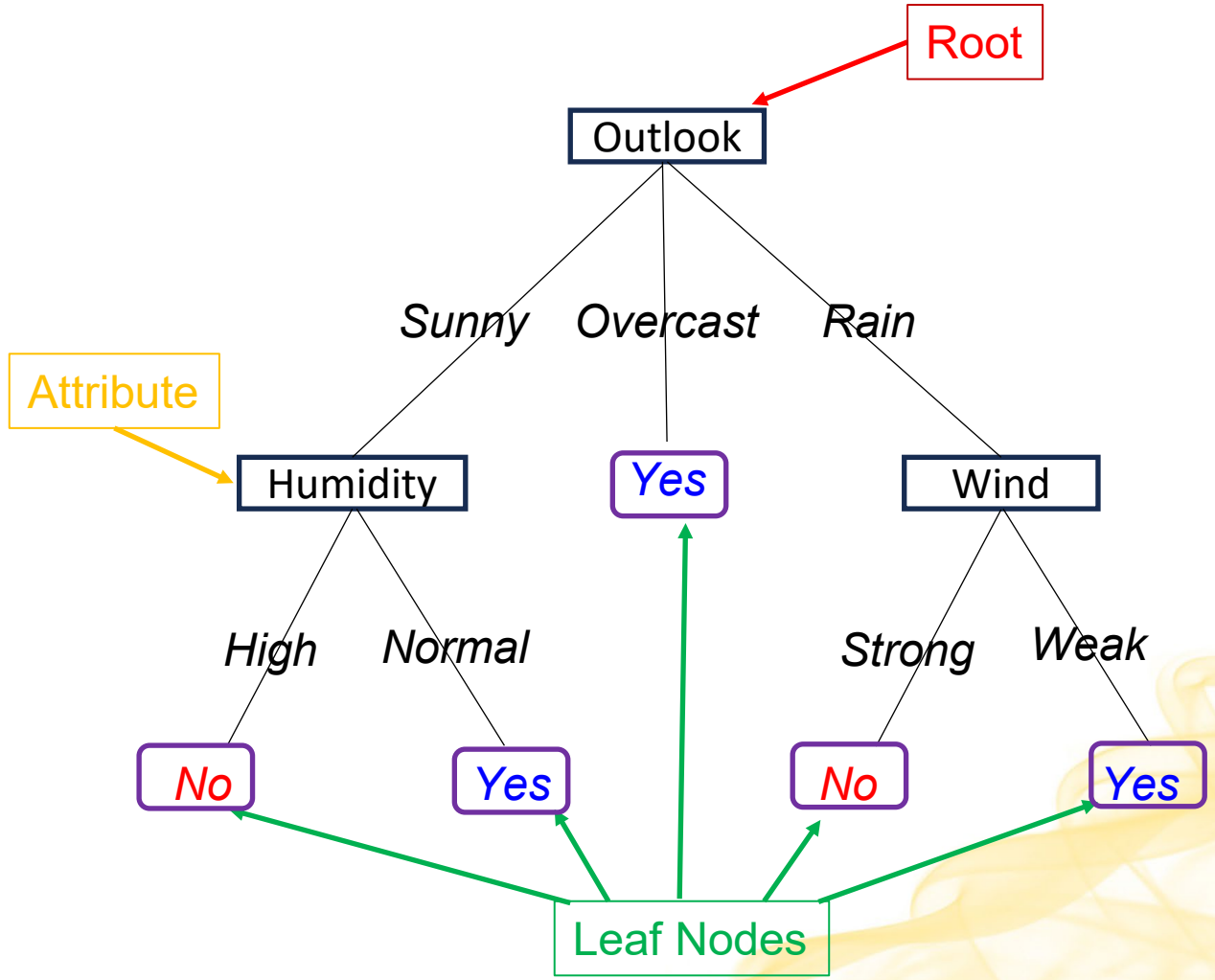
$$\text{Info. Gain} = 0.985 - \left(\frac{4}{7}0.811 + \frac{3}{7}0.918\right) = 0.128$$

Decision Trees

- Potential issues with decision trees
  - Overfitting – often happens when leaf nodes have only a few samples
  - Solutions
    - Pruning, cross-validation
- Available algorithms
  - ID3
  - C4.5

Decision Trees

(Outlook = Sunny $\wedge$ Humidity = Normal)
$\vee$ (Outlook = Overcast)
$\vee$ (Outlook = Rain $\wedge$ Wind = Weak)



Decision Trees

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No