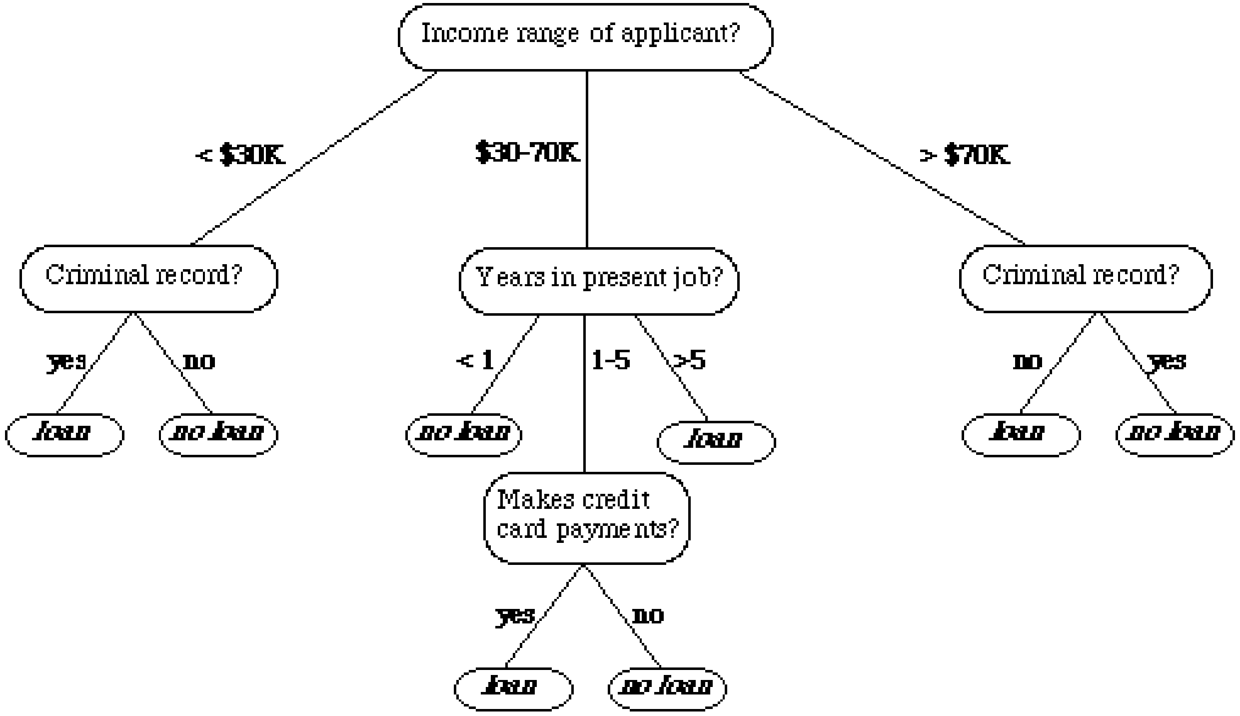
Classification Methods

- Logistic Regression
- Naïve Bayes
- k-Nearest Neighbors
- Decision Trees
- Support Vector Machines
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

Slide Credits : Vineeth N Balasubrahmanian

Bank loan

- what kind of income does the person have?
- How long have they held their current job?

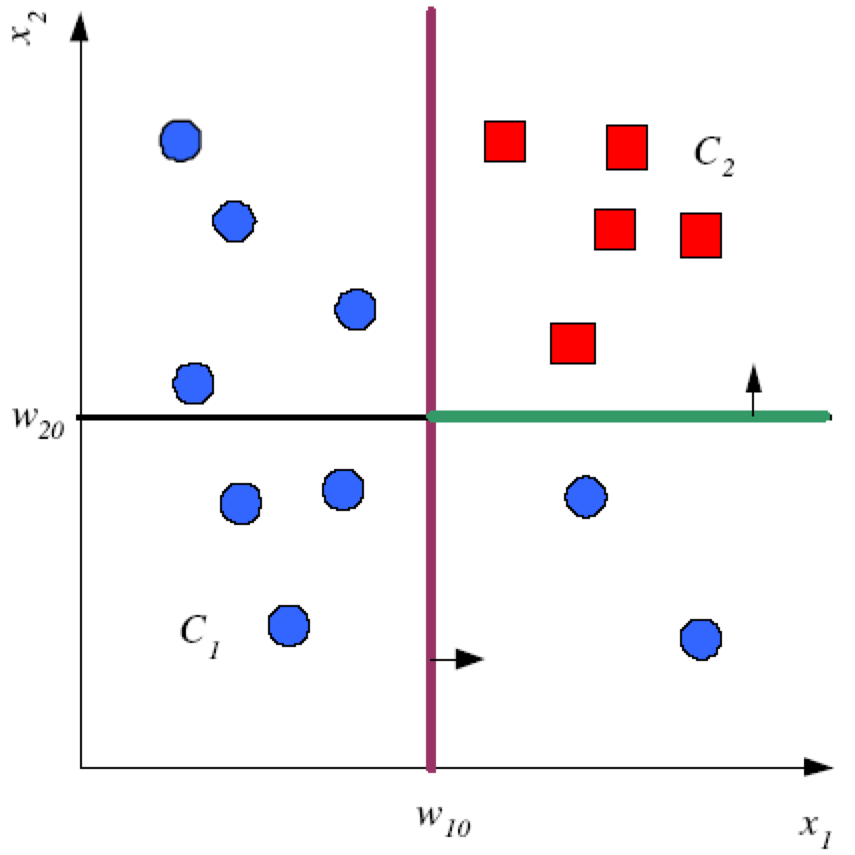


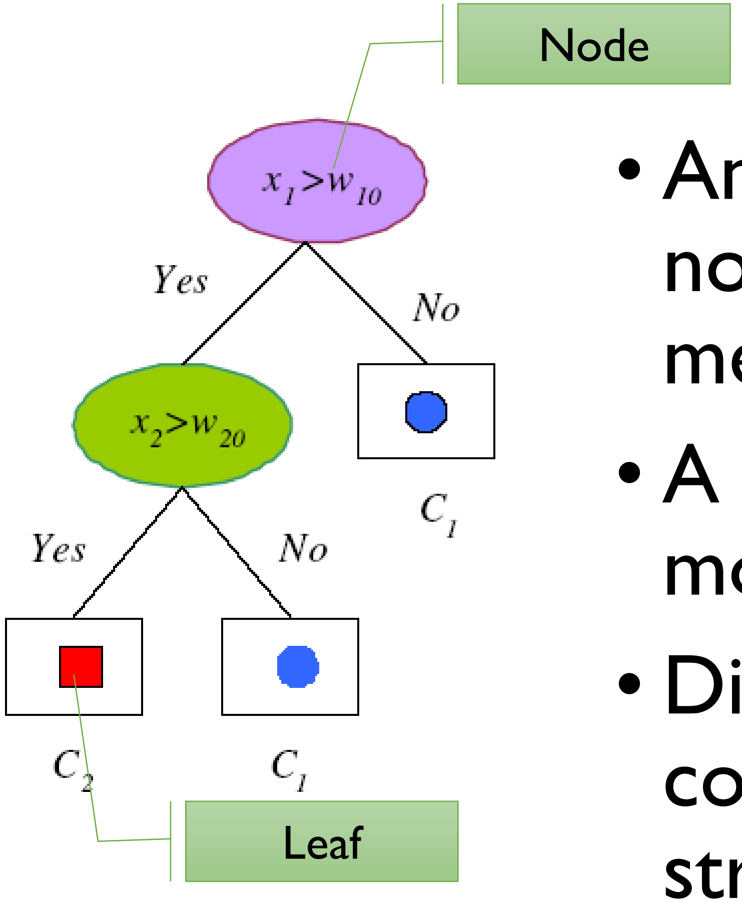
Example

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Trees



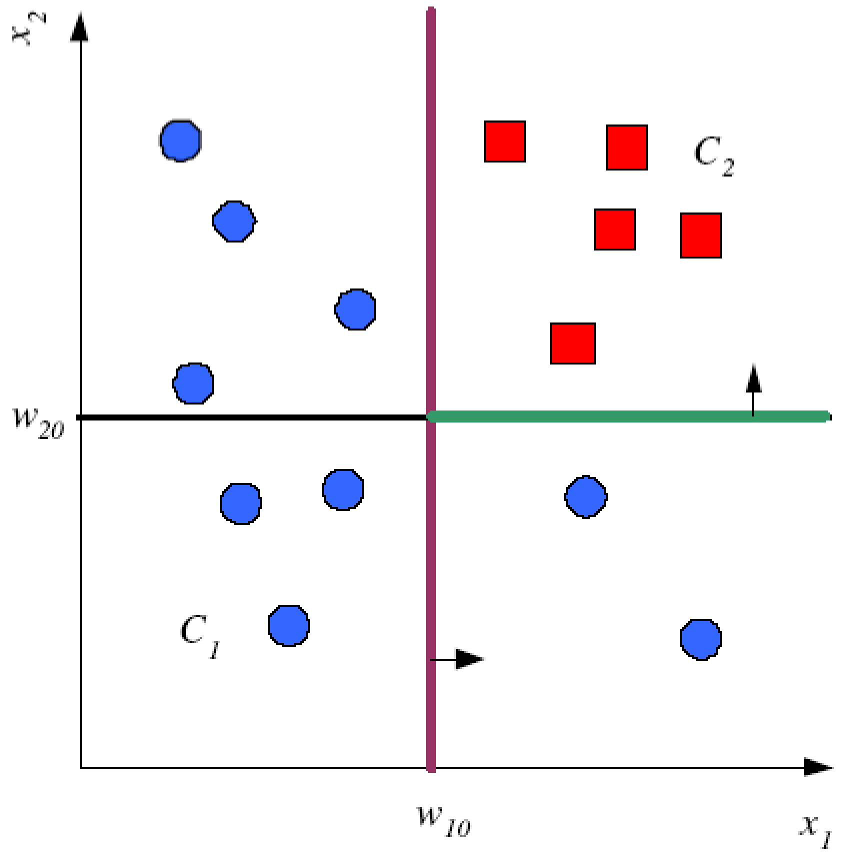


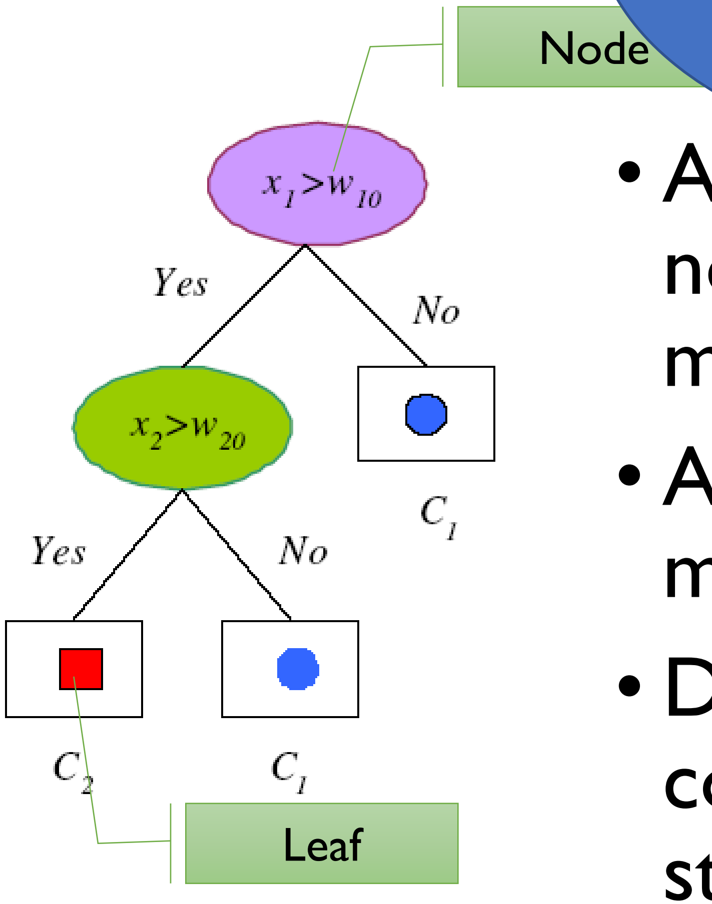
- An efficient nonparametric method
- A hierarchical model
- Divide-and–conquer strategy

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Decision Trees

A Non-parametric method means that there are no underlying assumptions about the distribution of the data.

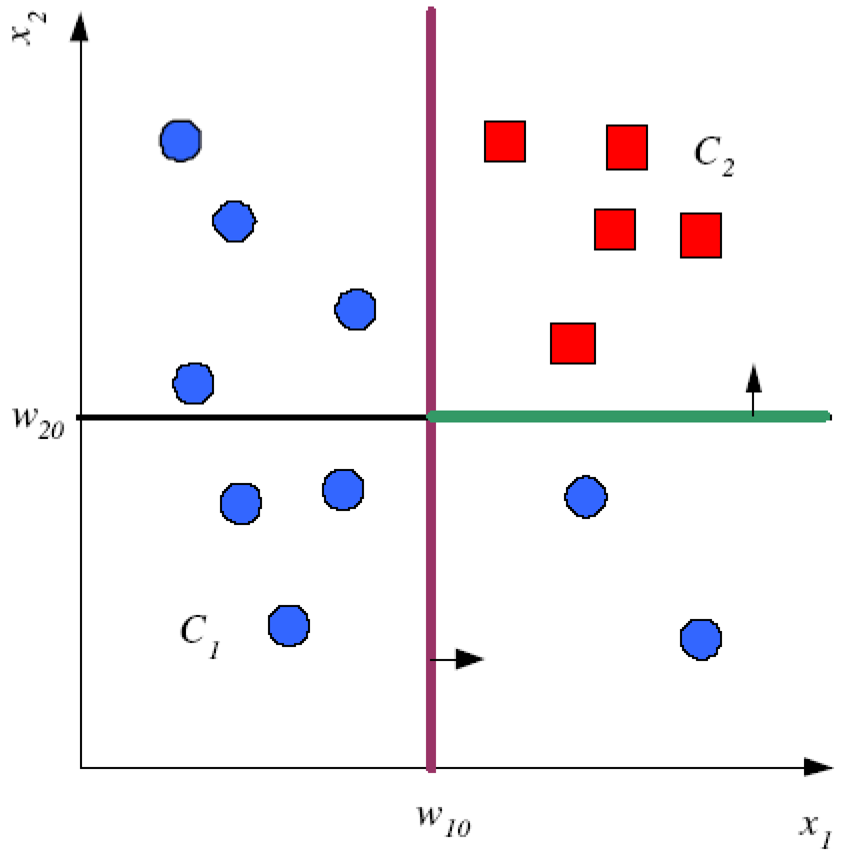


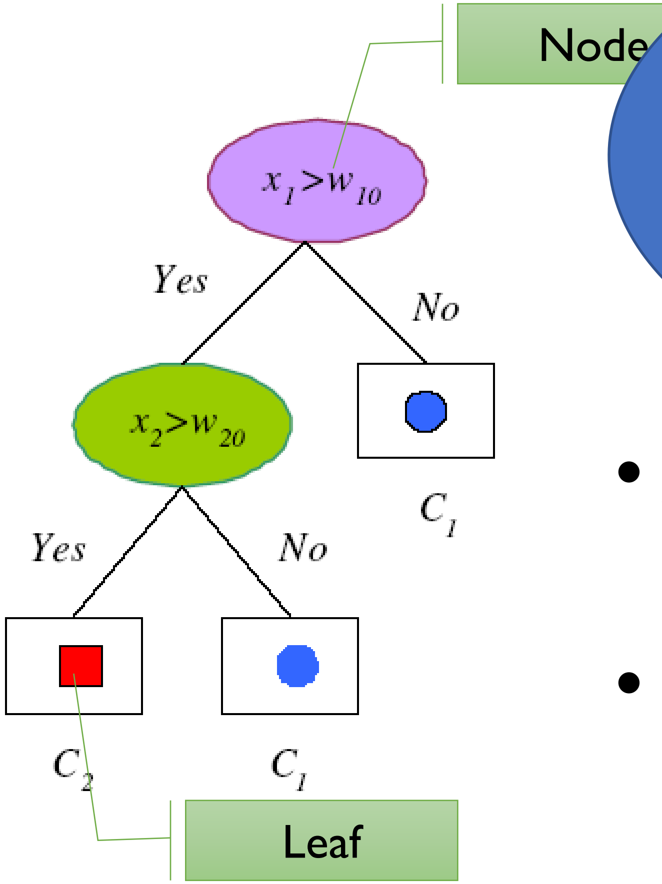


- An efficient nonparametric method
- A hierarchical model
- Divide-and–conquer strategy

Source: *Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)*

Decision Trees





Hierarchical means the model is defined by a series of questions that lead to a class label or a value when applied to any observation.

- meti
- A hierarchical model
- Divide-and–conquer strategy

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Divide and Conquer

- Internal decision nodes
  - Univariate: Uses a single attribute, x_i
    - Numeric x_i :
      - Binary split : $x_i > w_m$
    - Discrete x_i :
      - n-way split for n possible values
  - Multivariate: Uses more than one attributes, $\boldsymbol{x}$
- Leaves
  - Classification: Class labels, or proportions
  - Regression: Numeric; r average, or local fit
- Learning is greedy; find the best split recursively

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Classification Trees (C4.5, J48)

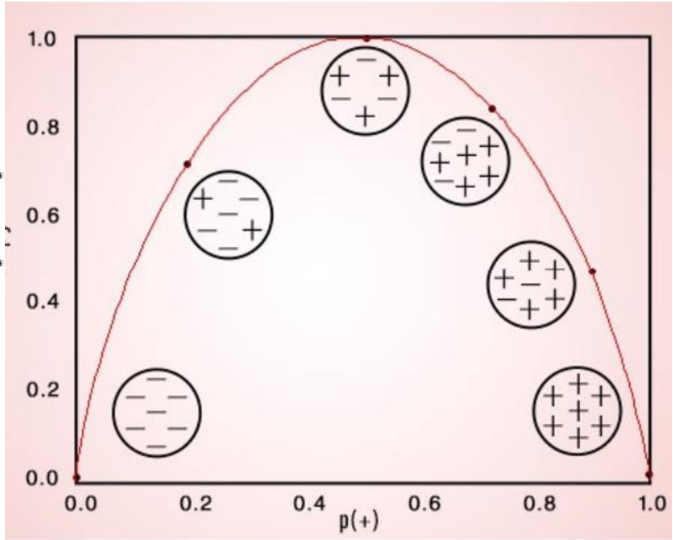
- For node *m*, N_m instances reach *m*, N^i_m belong to C_i

$$\hat{P}(C_i \mid \mathbf{x}, m) \equiv p^i_m = \frac{N^i_m}{N_m}$$

- Node *m* is **pure** if p^i_m is 0 or 1
- Measure of **impurity** is **entropy**

$$\mathcal{I}_m = -\sum_{i=1}^{K} p^i_m \log_2 p^i_m$$

2-class problem



Entropy in information theory specifies the average (expected) amount of information derived from observing an event

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Classification Trees

- If node m is pure, generate a leaf and stop, otherwise split and continue recursively
- **Impurity after split:** N_{mj} of N_m take branch j. N^i_{mj} belong to C_i

$$\hat{P}(C_i \mid \mathbf{x}, m, j) \equiv p^i_{mj} = \frac{N^i_{mj}}{N_{mj}}$$

$$\mathcal{I}'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p^i_{mj} \log_2 p^i_{mj}$$

- **Information Gain:** Expected reduction in impurity measure after split
- Choose the *best* attribute(s) (**with maximum information gain**) to split the remaining instances and make that attribute a decision node
  - You can use same logic to find best splitting value too

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Other Measures of Impurity

- The properties of functions measuring the **impurity** of a split:
  - $\phi(1/2, 1/2) \geq \phi(p, 1-p)$, for any $p \in [0,1]$
  - $\phi(0,1) = \phi(1,0) = 0$
  - $\phi(p, 1-p)$ is increasing in p on $[0, \frac{1}{2}]$

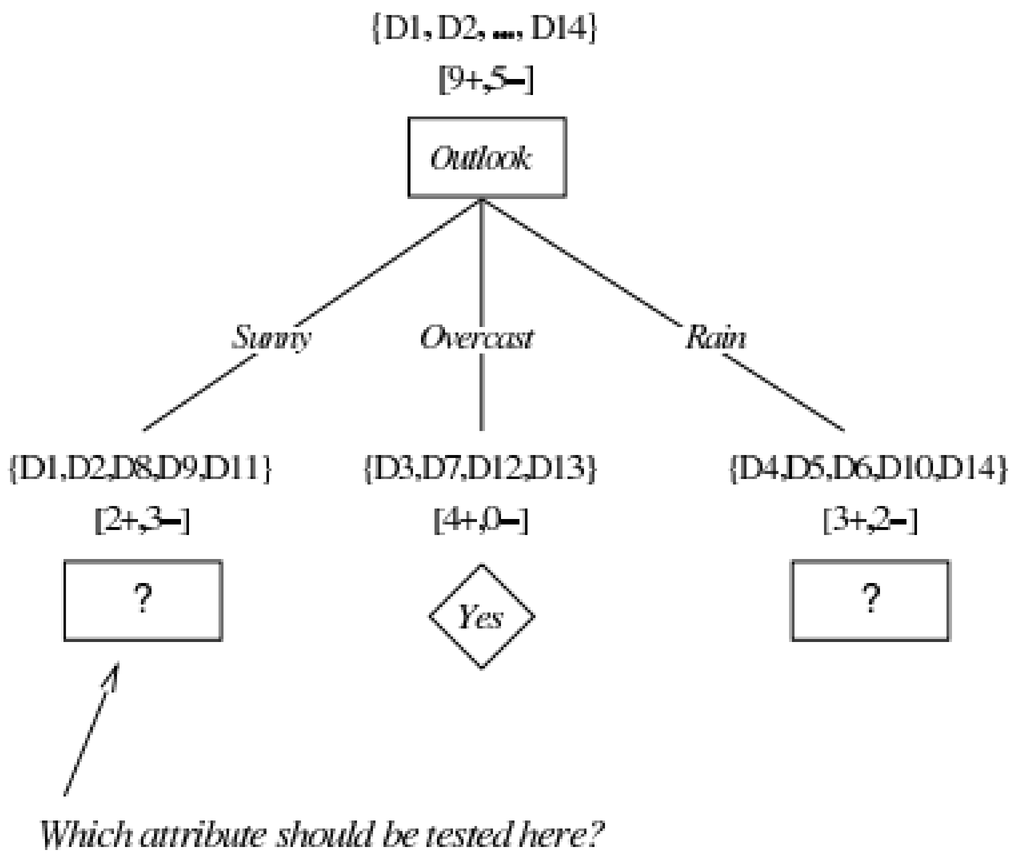
    and decreasing in p on $[\frac{1}{2}, 1]$
- Examples (other than entropy)
  - **Gini impurity/index:** $1 - \sum_{j=1}^{c} p_j^2$

Decision Trees: Example

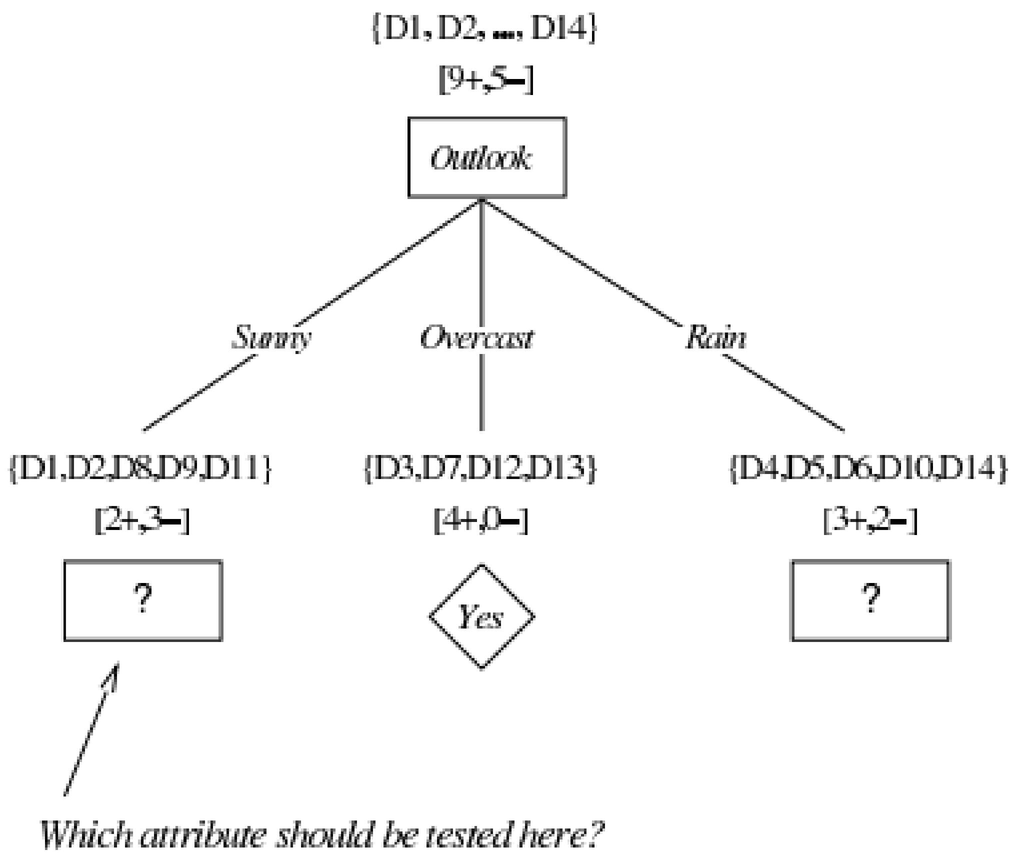
PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Trees: Example



Decision Trees: Example



$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain\,(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain\,(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain\,(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Overfitting and Generalization

- Overfitting can occur with noisy training examples, also when small numbers of examples are associated with leaf nodes. How to handle?
- **Pruning:** Remove subtrees for better generalization (decrease variance)
  - Prepruning: Early stopping
  - Postpruning: Grow the whole tree then prune subtrees which overfit on the pruning set
  - Prepruning is faster, postpruning is more accurate

Overfitting and Generalization

- **Occam's Razor principle:** when multiple hypotheses can solve a problem, choose the simplest one

- How to select "best" tree:
  - Measure performance over separate validation data set
  - **Minimum Description Length:** Minimize `size(tree) + size(misclassifications(tree))`

Pruning based on validation data

- Reduced-error pruning, is to consider each of the decision nodes in the tree to be candidates for pruning
- Pruning a decision node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with that node
- Nodes are removed only if the resulting pruned tree performs no worse than-the original over the validation set.
- Reduced error pruning has the effect that any leaf node added due to coincidental regularities in the training set is likely to be pruned because these same coincidences are unlikely to occur in the validation set

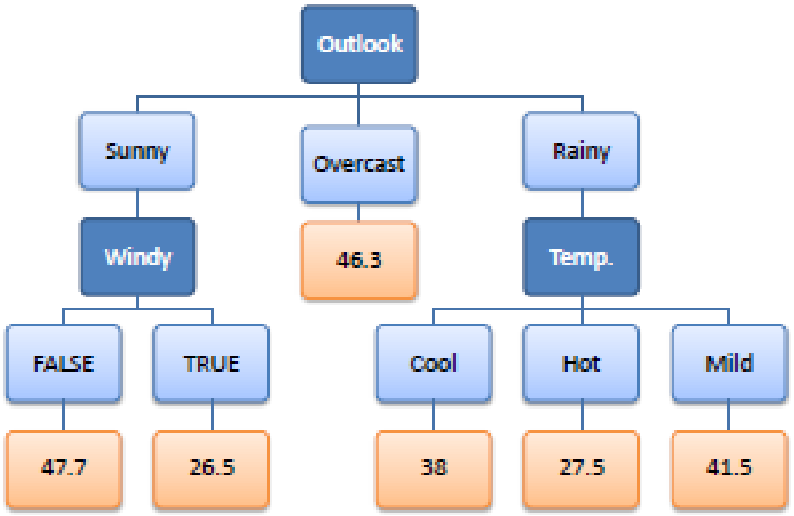
Decision Trees for Regression

- Entropy as a measure of impurity is a useful criteria for classification.
- impurity measure using the **weighted mean squared error** (**MSE**) of the children nodes or variance

$$\mathrm{MSE}(t) = \frac{1}{N_t} \sum_{i \in D_t} (y^{(i)} - \hat{y}_t)^2 \qquad \hat{y}_t = \frac{1}{N_t} \sum_{i \in D_t} y^{(i)}$$

- Attribute chosen so that it reduces variance od standard deviation

Predictors				Target
Outlook	**Temp.**	**Humidity**	**Windy**	**Hours Played**
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	46
Sunny	Mild	High	False	45
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	46
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$\boldsymbol{Count} = n = 14$$

$$\boldsymbol{Average} = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$\boldsymbol{Standard\ Deviation} = S = \sqrt{\frac{\sum (x - \bar{x})^2}{n}} = 9.32$$

$$\boldsymbol{Coeffeicient\ of\ Variation} = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

$$S(T,X)=\sum_{c\in X}P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

$$SDR(T,X)=S(T)-S(T,X)$$

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Multivariate Trees