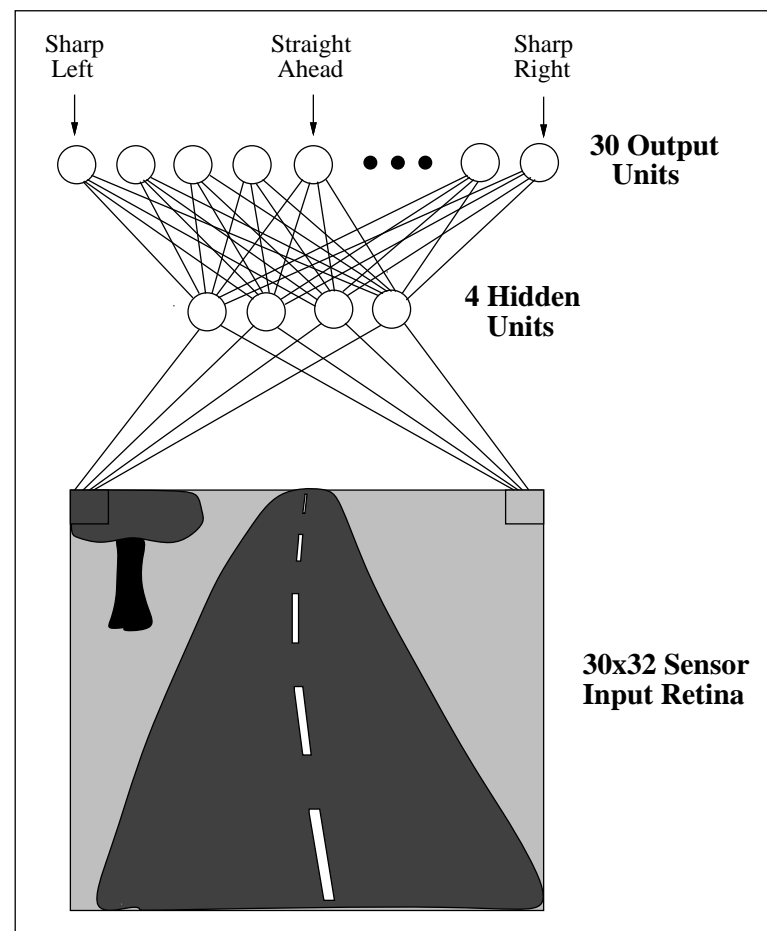


# SZTUCZNA INTELIGENCJA I SYSTEMY DORADCZE

## UCZENIE MASZYNOWE - WPROWADZENIE I DRZEWA DECYZYJNE

# System ALVINN

ALVINN prowadzi samochód po autostradzie z szybkością 70 mil na godzinę



www.wisewire.com, adaptuje sie do uzytkownika



# Uczenie indukcyjne

**Obiekty:** dane reprezentujące rzeczywisty stan lub obiekt, tworzą przestrzeń obiektów  $X$

**Decyzja:** Funkcja  $dec : X \rightarrow V_{dec}$  przypisująca obiektom wartość decyzji z ustalonego zbioru  $V_{dec}$

**Zbiór przykładów:** ustalony zbiór obiektów z  $X$  z przypisanymi wartościami decyzji:  
 $(x_1, dec(x_1)), \dots, (x_m, dec(x_m))$

**Problem:**

Z danego zbioru przykładów nauczyć się funkcję (hipotezę)  $h : X \rightarrow V_{dec}$  aproksymującą decyzję  $dec$  tak, żeby możliwie **najbardziej poprawnie** przypisywała ją obiektom z przestrzeni  $X$ , dla których nieznana jest wartość decyzji  $dec$

## Uczenie indukcyjne: kolko i krzyzyk

Obiekty to pary: stan gry i ocena stanu

<i>O</i>	<i>O</i>	<i>X</i>
	<i>X</i>	
<i>X</i>		

, +1

Problem polega na nauczeniu się właściwej oceny stanu gry

## Uczenie indukcyjne: uprawianie sportu

Obiekty to wektory wartości opisujące bieżące warunki pogodowe

Przykład	Atrybuty						Decyzja
	<i>Niebo</i>	<i>Temp</i>	<i>Wilgotn</i>	<i>Wiatr</i>	<i>Woda</i>	<i>Prognoza</i>	<i>Sport</i>
<i>Dzień 1</i>	<i>Słońce</i>	<i>Ciepło</i>	<i>Normalna</i>	<i>Silny</i>	<i>Ciepła</i>	<i>Bez zmian</i>	<i>Tak</i>
<i>Dzień 2</i>	<i>Słońce</i>	<i>Ciepło</i>	<i>Wysoka</i>	<i>Silny</i>	<i>Ciepła</i>	<i>Bez zmian</i>	<i>Tak</i>
<i>Dzień 3</i>	<i>Deszcz</i>	<i>Zimno</i>	<i>Normalna</i>	<i>Silny</i>	<i>Ciepła</i>	<i>Zmiana</i>	<i>Nie</i>
<i>Dzień 4</i>	<i>Słońce</i>	<i>Ciepło</i>	<i>Wysoka</i>	<i>Silny</i>	<i>Chłodna</i>	<i>Zmiana</i>	<i>Tak</i>

Problem polega na nauczaniu się podejmowania właściwego wyboru na podstawie bieżących warunków pogodowych

## Rodzaje decyzji

Decyzja może przyjmować wartości:

- ◇ rzeczywiste (decyzja ciągła)
- ◇ dyskretne
- ◇ binarne (TRUE lub FALSE)

## Hipoteza spojna

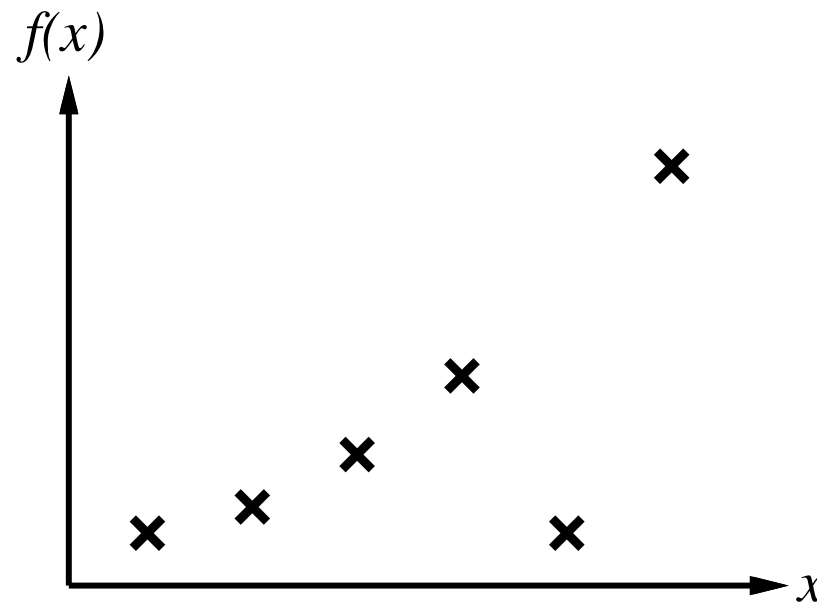
Hipoteza  $H$  jest **spójna** na zbiorze treningowym  $(x_1, dec(x_1)), \dots, (x_m, dec(x_m))$ , jeśli

$$h(x_i) = dec(x_i) \text{ dla każdego } 1 \leq i \leq m$$



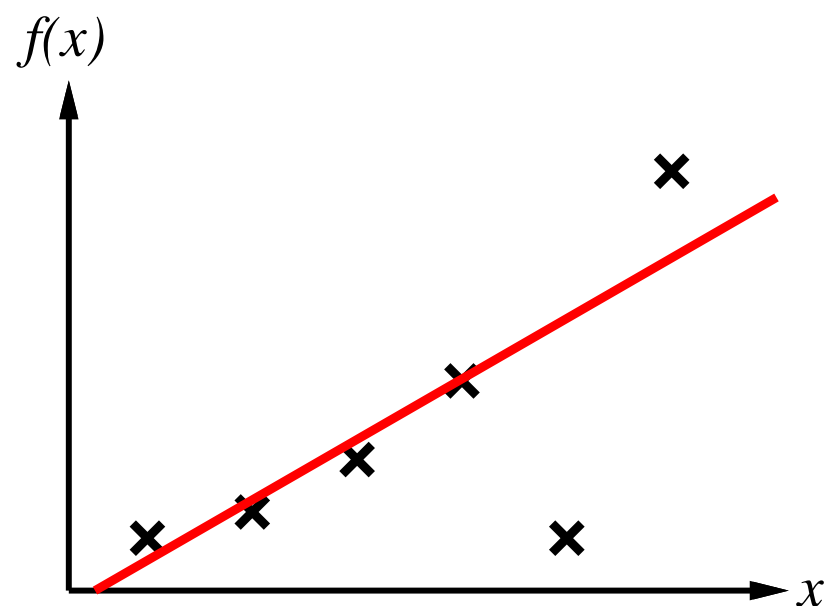
# Regresja (decyzja ciągła)

Regresja to aproksymacja decyzji o wartości ciągłej



# Regresja (decyzja ciągła)

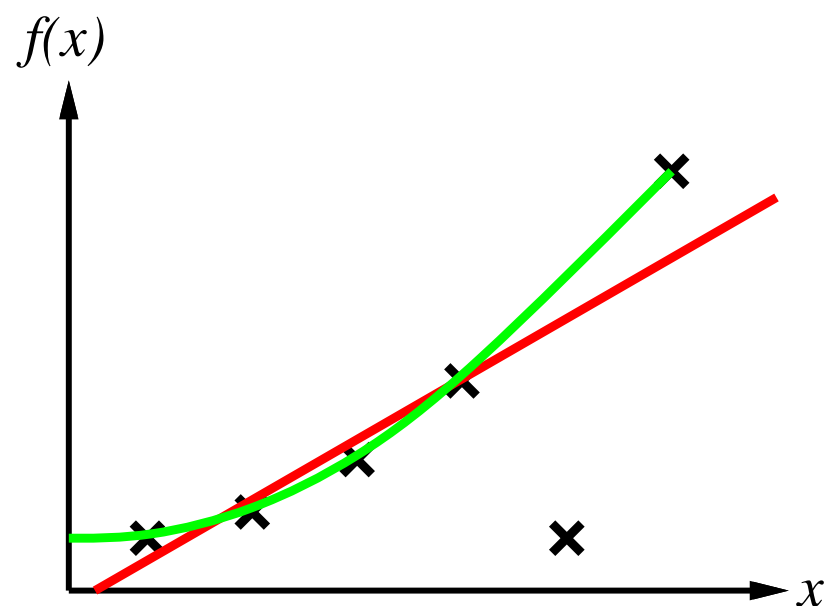
Regresja to aproksymacja decyzji o wartości ciągłej



Regresja liniowa

# Regresja (decyzja ciągła)

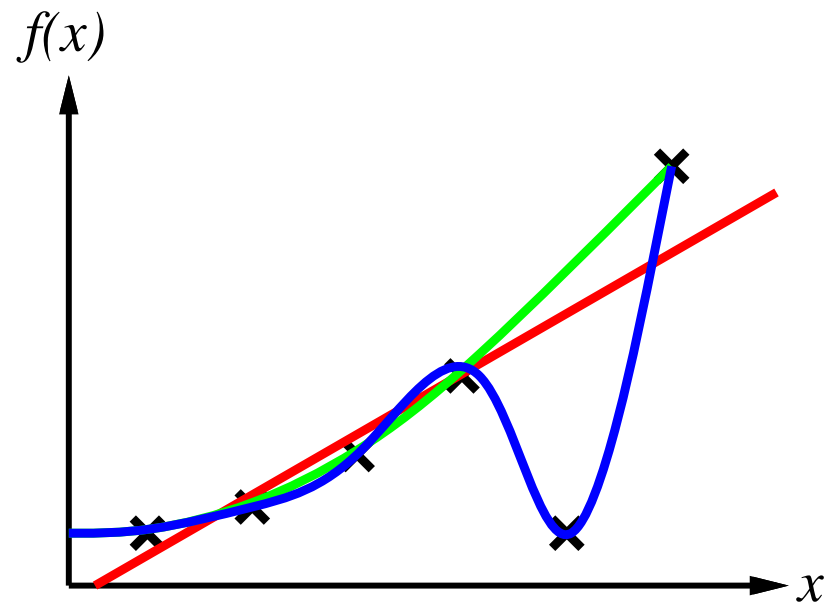
Regresja to aproksymacja decyzji o wartości ciągłej



Regresja kwadratowa

# Regresja (decyzja ciągła)

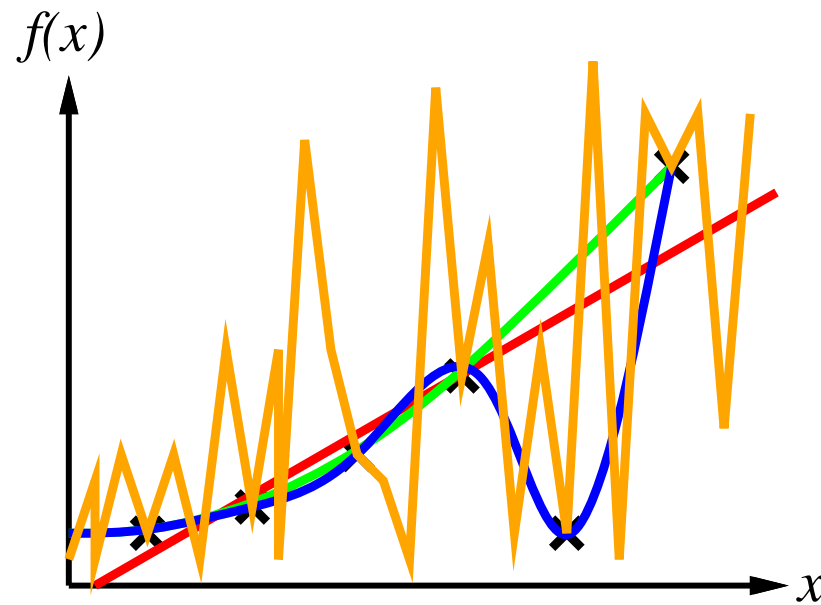
Regresja to aproksymacja decyzji o wartości ciągłej



Regresja wyższego stopnia (spójna)

# Regresja (decyzja ciągła)

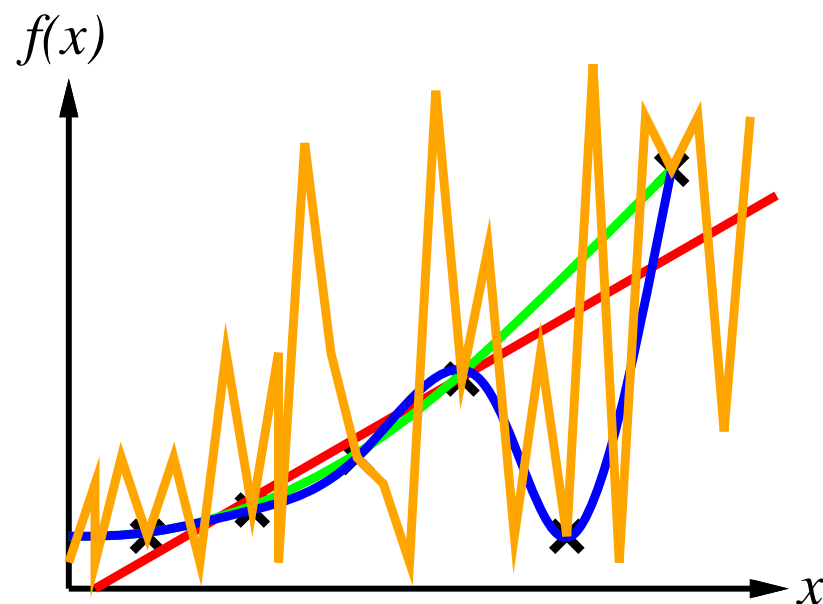
Regresja to aproksymacja decyzji o wartości ciągłej



Regresja kawałkami liniowa (spójna)

# Brzytwa Ockhama

Brzytwa Ockhama wybiera hipotezę, która maksymalizuje połączenie spójności i prostoty



Wybór aproksymacji zależy od funkcji oceny spójności i prostoty

## Opis obiektów przez atrybuty

**Atrybuty:** zbiór atrybutów  $A = \{a_1, a_2, \dots, a_n\}$

**Obiekty:** Wektory wartości atrybutów  $x = \langle x_1, x_2, \dots, x_n \rangle$

Wartości atrybutów mogą być:

- ◇ ciągłe
- ◇ dyskretne
- ◇ binarne (TRUE lub FALSE)

## Opis obiektów przez atrybuty: przykład

Decyzja na grę w tenisa jest binarna: Yes lub No

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



# Przestrzen hipotez

Ile jest różnych hipotez (funkcji) binarnych dla  $n$  atrybutów binarnych??

## Przestrzen hipotez

Ile jest różnych hipotez (funkcji) binarnych dla  $n$  atrybutów binarnych??

= liczba funkcji binarnych dla dziedziny z  $2^n$  obiektami =  $2^{2^n}$

## Przestrzen hipotez

Ile jest różnych hipotez (funkcji) binarnych dla  $n$  atrybutów binarnych??

= liczba funkcji binarnych dla dziedziny z  $2^n$  obiektami =  $2^{2^n}$

np. dla 6 atrybutów binarnych jest 18,446,744,073,709,551,616 hipotez

## Ograniczanie przestrzeni hipotez

Przestrzeń hipotez można ograniczyć do ustalonej klasy hipotez

## Ograniczanie przestrzeni hipotez: przykład

Przestrzeń hipotez można ograniczyć do ustalonej klasy hipotez

Ile jest czysto koniunkcyjnych funkcji (np.  $Hungry \wedge \neg Rain$ )??

## Ograniczanie przestrzeni hipotez: przykład

Przestrzeń hipotez można ograniczyć do ustalonej klasy hipotez

Ile jest czysto koniunkcyjnych funkcji (np.  $Hungry \wedge \neg Rain$ )??

Każdy atrybut może wystąpić jako literał **pozytywny**,  
jako literał **negatywny** lub wcale

## Ograniczanie przestrzeni hipotez: przykład

Przestrzeń hipotez można ograniczyć do ustalonej klasy hipotez

Ile jest czysto koniunkcyjnych funkcji (np.  $Hungry \wedge \neg Rain$ )??

Każdy atrybut może wystąpić jako literał **pozytywny**,  
jako literał **negatywny** lub wcale

$\Rightarrow 3^n$  różnych funkcji koniunkcyjnych

# Zwiększanie przestrzeni hipotez

Co powoduje zwiększenie klasy dopuszczalnych hipotez??



## Zwiększanie przestrzeni hipotez

Co powoduje zwiększenie klasy dopuszczalnych hipotez??

– zwiększa szansę, że funkcja docelowa może być wyrażona

## Zwiększanie przestrzeni hipotez

Co powoduje zwiększenie klasy dopuszczalnych hipotez??

- zwiększa szansę, że funkcja docelowa może być wyrażona
- zwiększa liczbę hipotez zgodnych ze zbiorem treningowym

## Zwiększanie przestrzeni hipotez

Co powoduje zwiększenie klasy dopuszczalnych hipotez??

- zwiększa szansę, że funkcja docelowa może być wyrażona
- zwiększa liczbę hipotez zgodnych ze zbiorem treningowym
- ⇒ może spowodować gorszą skuteczność predykcji

## Empiryczna miara jakości hipotezy

Dane dzielone są na **zbiór treningowy**  $U_{trn}$  i **zbiór testowy**  $U_{tst}$

Hipoteza  $h : X \rightarrow V_{dec}$  jest indukowana na podstawie zbioru treningowego  $U_{trn}$

**Skuteczność** hipotezy  $Accuracy(h)$  jest mierzona proporcją poprawnie sklasyfikowanych obiektów ze zbioru testowego

$$Accuracy(h) = \frac{|\{x \in U_{tst} : h(x) = dec(x)\}|}{|U_{tst}|}$$

# Metody wnioskowania

Metody wnioskowania dla danych opisanych przez zbiór atrybutów z decyzją dyskretną:

- ◇ Drzewa decyzyjne
- ◇ Systemy regułowe
- ◇ Sieci neuronowe
- ◇ Wnioskowanie oparte na podobieństwie
- ◇ Sieci bayessowskie

# Drzewa decyzyjne: reprezentacja

## Węzły wewnętrzne

Każdy związany z jednym atrybutem,  
reprezentuje test wartości tego atrybutu

## Gałęzie

Każda związana z jedną wartością lub z podzbiorem wartości  
atrybutu węzła, z którego wychodzi gałąź,  
odpowiada obiektom danych z pasującymi wartościami atrybutu

## Liście

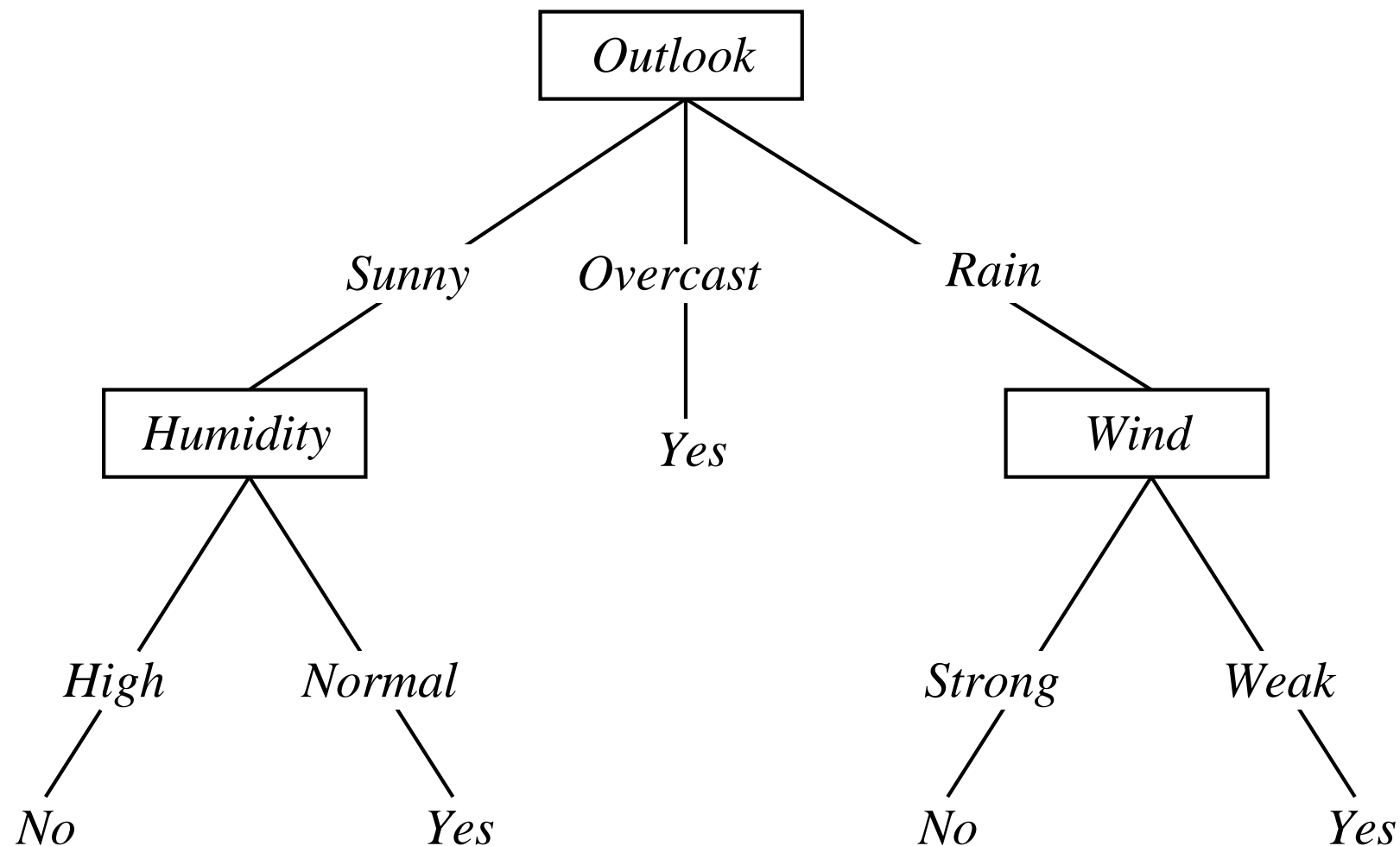
Każdy związany z decyzją lub rozkładem decyzji, odpowiada obiektom  
danych pasującym do ścieżki prowadzącej do danego liścia

# Drzewa decyzyjne: przykład

Reprezentacja danych

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

## Drzewa decyzyjne: przykład





# Drzewa decyzyjne: moc wyrażania

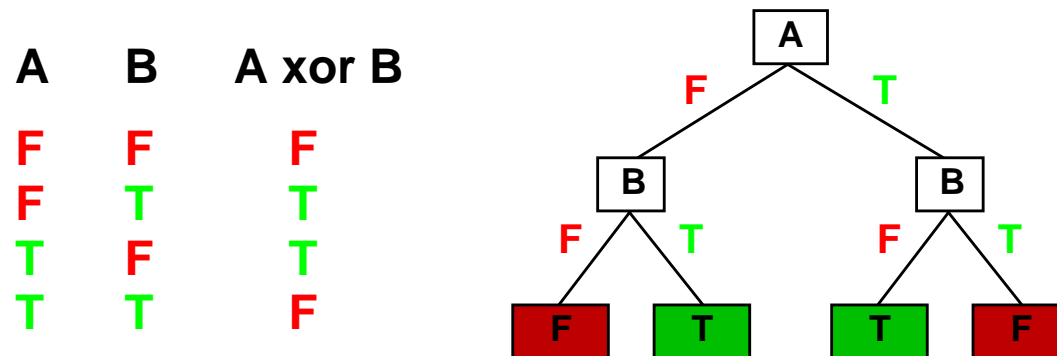
**Fakt 1:** Dla każdego zbioru treningowego istnieje spójne drzewo decyzyjne

**Dowód:** zaczynamy od pustego drzewa i dla każdego obiektu danych dokładamy

ścieżkę prowadzącą przez wszystkie atrybuty z wartościami krawędzi odpowiadającymi wartościom atrybutów w obiekcie

**Fakt 2:** Dla każdej funkcji istnieje spójne drzewo decyzyjne

**Dowód:** można utworzyć zbiór treningowy zawierający obiekty odpowiadające wszystkim kombinacjom wartości atrybutów o decyzji zgodnej z daną funkcją



# Drzewa decyzyjne: trenowanie

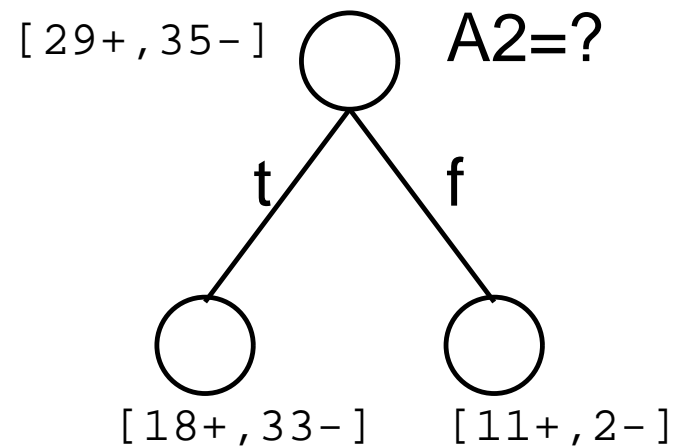
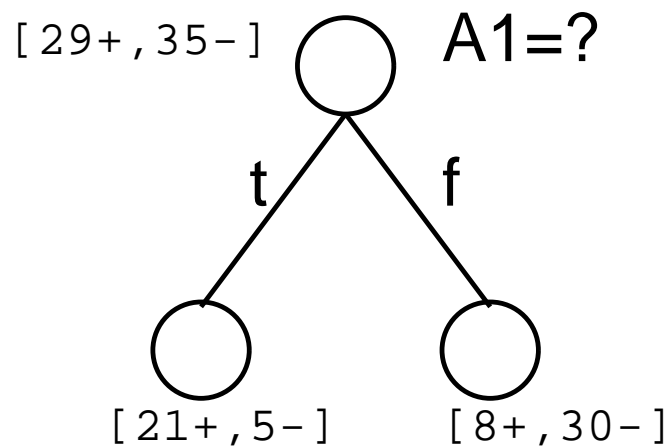
**Cel:** znalezienie drzewa spójnego ze zbiorem treningowym

**Pomysł:** rekurencyjne wybieranie **najbardziej znaczącego** atrybutu jako korzeń poddrzewa

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi ← {elements of examples with best =  $v_i$ }
      subtree ← DTL(examplesi, attributes – best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
  return tree
```

## Drzewa decyzyjne: wybór atrybutu

Różne atrybuty dają różne rozkłady decyzji w gałęziach



Funkcja CHOOSE-ATTRIBUTE wybiera najlepszy z nich

# Entropia

Dany jest rozkład prawdopodobieństwa  $\langle p_1, \dots, p_n \rangle$

Miara **entropii** wyznacza, ile informacji niesie z sobą ten rozkład

$$Entropy(\langle p_1, \dots, p_n \rangle) = \sum_{i=1}^n -p_i \log_2 p_i$$

# Entropia

Dany jest rozkład prawdopodobieństwa  $\langle p_1, \dots, p_n \rangle$

Miara **entropii** wyznacza, ile informacji niesie z sobą ten rozkład

$$Entropy(\langle p_1, \dots, p_n \rangle) = \sum_{i=1}^n -p_i \log_2 p_i$$

$S$  — zbiór obiektów danych

$S_d$  — zbiór obiektów w  $S$  z decyzją  $d$

$$Entropy(S) = \sum_{d \in V_{dec}} \frac{|S_d|}{|S|} \log_2 \frac{|S|}{|S_d|}$$

Entropia = średnia liczba bitów potrzebna do zakodowania decyzji  $d$  dla losowo wybranego obiektu ze zbioru  $S$  (przy optymalnym kodowaniu decyzji)

Dlaczego??

# Entropia

Dany jest rozkład prawdopodobieństwa  $\langle p_1, \dots, p_n \rangle$

Miara **entropii** wyznacza, ile informacji niesie z sobą ten rozkład

$$Entropy(\langle p_1, \dots, p_n \rangle) = \sum_{i=1}^n -p_i \log_2 p_i$$

$S$  — zbiór obiektów danych

$S_d$  — zbiór obiektów w  $S$  z decyzją  $d$

$$Entropy(S) = \sum_{d \in V_{dec}} -\frac{|S_d|}{|S|} \log_2 \frac{|S_d|}{|S|}$$

Entropia = średnia liczba bitów potrzebna do zakodowania decyzji  $d$  dla losowo wybranego obiektu ze zbioru  $S$  (przy optymalnym kodowaniu decyzji)

Dlaczego??

Optymalne kodowanie przydziela  $-\log_2 p$  bitów do decyzji występującej z prawdopodobieństwem  $p$

## Entropia: dwie decyzje

Dane są dwie decyzje: pozytywna ( $\oplus$ ) i negatywna ( $\ominus$ )

$p_{\oplus} = \frac{|S_{\oplus}|}{|S|}$  — proporcja obiektów z decyzją pozytywną w zbiorze  $S$

$p_{\ominus} = \frac{|S_{\ominus}|}{|S|}$  — proporcja obiektów z decyzją negatywną w zbiorze  $S$

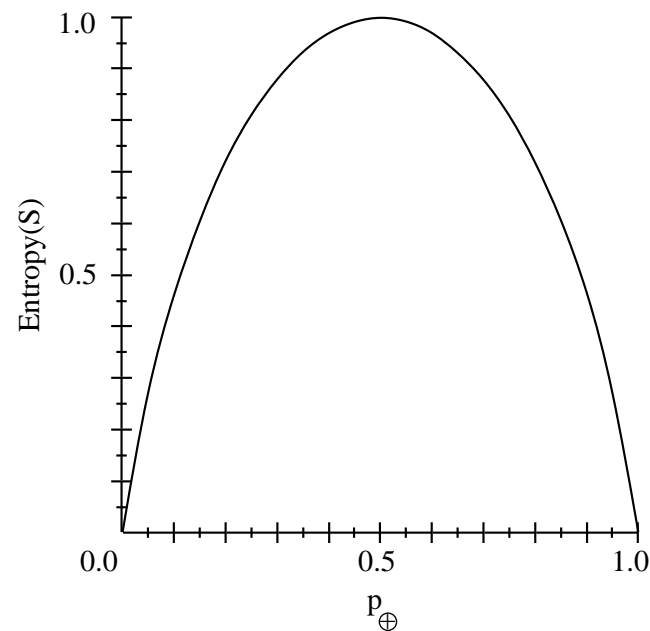
## Entropia: dwie decyzje

Dane są dwie decyzje: pozytywna ( $\oplus$ ) i negatywna ( $\ominus$ )

$p_{\oplus} = \frac{|S_{\oplus}|}{|S|}$  — proporcja obiektów z decyzją pozytywną w zbiorze  $S$

$p_{\ominus} = \frac{|S_{\ominus}|}{|S|}$  — proporcja obiektów z decyzją negatywną w zbiorze  $S$

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$





## Zysk informacji dla atrybutu symbolicznego

Zysk informacji  $Gain(S, a)$

= redukcja entropii przy podziale zbioru względem atrybutu  $a$

$S_v$  — zbiór obiektów w  $S$  z wartością atrybutu  $a = v$

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

# Zysk informacji dla atrybutu symbolicznego

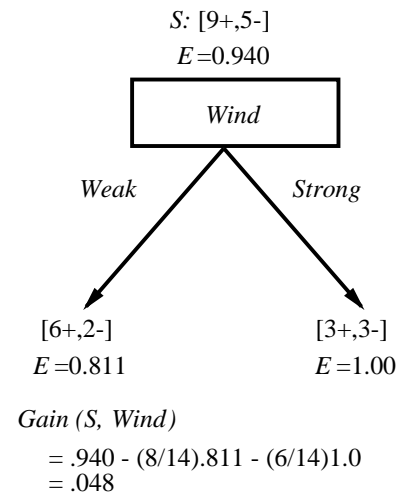
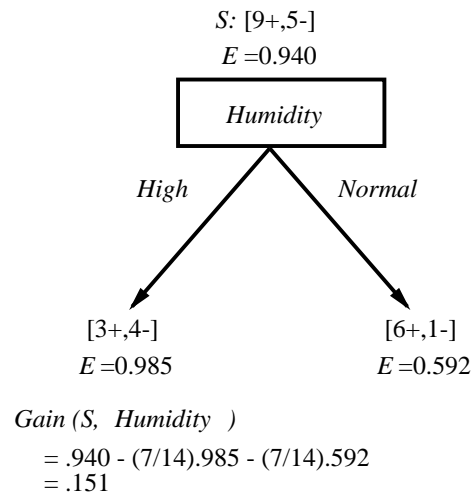
Zysk informacji  $Gain(S, a)$

= redukcja entropii przy podziale zbioru względem atrybutu  $a$

$S_v$  — zbiór obiektów w  $S$  z wartością atrybutu  $a = v$

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

**Which attribute is the best classifier?**



## Wybor atrybutu

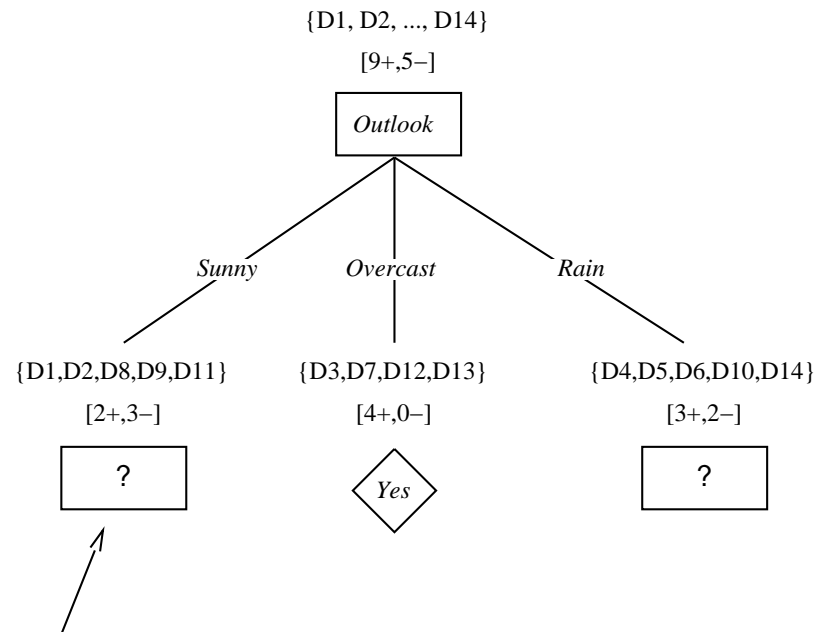
Drzewo decyzyjne w każdym węźle wybiera atrybut  $a$  z największym zyskiem informacji, tzn. z największą wartością  $Gain(S, a)$

## Wybor atrybutu: przykład

Zbiór danych

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

# Wybor atrybutu: przykład



*Which attribute should be tested here?*

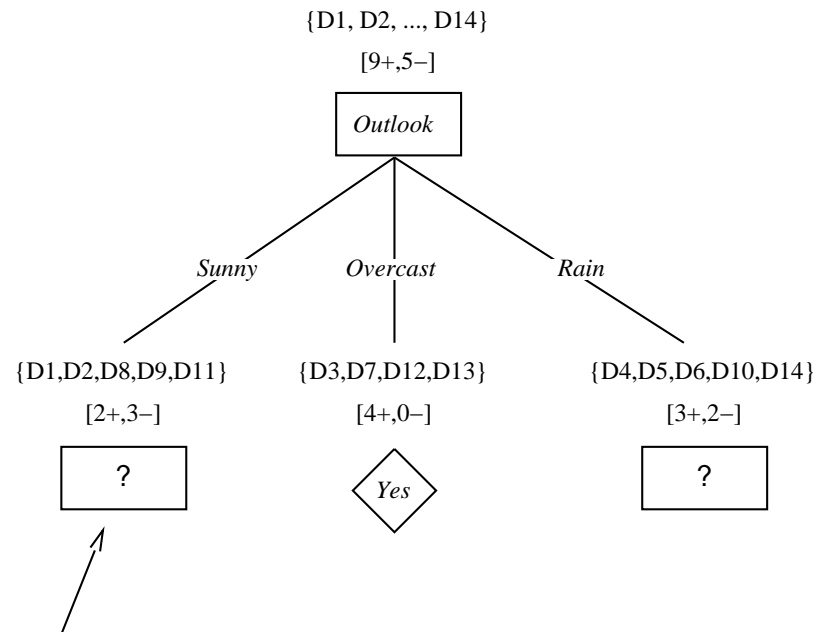
$$S_{\text{sunny}} = \{D1, D2, D8, D9, D11\}$$

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

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

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

# Wybor atrybutu: przykład



Which attribute should be tested here?

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

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

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

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

Najlepszym atrybutem jest *Humidity*

## Zysk informacji dla atrybutu numerycznego

Zysk informacji  $Gain(S, a, c)$

= redukcja entropii względem cięcia binarnego  $c$  na atrybucie  $a$

$c$  — wartość cięcia

$s_{a < c}$  — zbiór obiektów z wartościami atrybutu  $a$  poniżej cięcia

$s_{a \geq c}$  — zbiór obiektów z wartościami atrybutu  $a$  powyżej cięcia

$$Gain(S, a, c) = Entropy(S) - \frac{|s_{a < c}|}{|S|} Entropy(s_{a < c}) - \frac{|s_{a \geq c}|}{|S|} Entropy(s_{a \geq c})$$

## Wybor ciecia

Drzewo decyzyjne wybiera atrybut rozpatrując **najlepsze cięcia** dla atrybutów numerycznych



## Wybor ciecia

Drzewo decyzyjne wybiera atrybut rozpatrując **najlepsze cięcia** dla atrybutów numerycznych

### Przykład

Dany jest atrybut numeryczny *Temperature* z następującymi wartościami w zbiorze treningowym:

<i>Temperature</i>	4	8	16	22	26	32
<i>PlayTennis</i>	No	No	Yes	Yes	Yes	No

Najlepsze cięcie??

## Wybor ciecia

Drzewo decyzyjne wybiera atrybut rozpatrując **najlepsze cięcia** dla atrybutów numerycznych

### Przykład

Dany jest atrybut numeryczny *Temperature* z następującymi wartościami w zbiorze treningowym:

<i>Temperature</i>	4	8	16	22	26	32
<i>PlayTennis</i>	No	No	Yes	Yes	Yes	No

### Najlepsze cięcie??

$$Gain(S, a, 0) = 1.0 - (0/6)0.0 - (6/6)1.0 = 0.0$$

$$Gain(S, a, 12) = 1.0 - (2/6)0.0 - (4/6)0.811 = 0.439$$

$$Gain(S, a, 29) = 1.0 - (5/6)0.971 - (1/6)0.0 = 0.191$$

Najlepsze cięcie:  $c = 12$

## Wielokrotny wybór atrybutu

Ten sam atrybut **numeryczny** może być wybrany **kilkakrotnie** na jednej ścieżce od korzenia do liścia:

**Przykład:** najpierw cięcie  $c_1 = 12$ , potem  $c_2 = 29$

<i>Temperature</i>	4	8	16	22	26	32
<i>PlayTennis</i>	No	No	Yes	Yes	Yes	No

Ale każdy atrybut **symboliczny** może być wybrany **co najwyżej raz** !

## Wybor atrybutu: normalizacja

### Problem

Miara *Gain* faworyzuje atrybuty z wieloma wartościami, w szczególności atrybut z wartościami jednoznacznymi, np. PESEL, ma maksymalną wartość *Gain*, ale jego użyteczność dla nowych przykładów jest żadna

## Wybor atrybutu: normalizacja

### Problem

Miara *Gain* faworyzuje atrybuty z wieloma wartościami, w szczególności atrybut z wartościami jednoznacznymi, np. PESEL, ma maksymalną wartość *Gain*, ale jego użyteczność dla nowych przykładów jest żadna

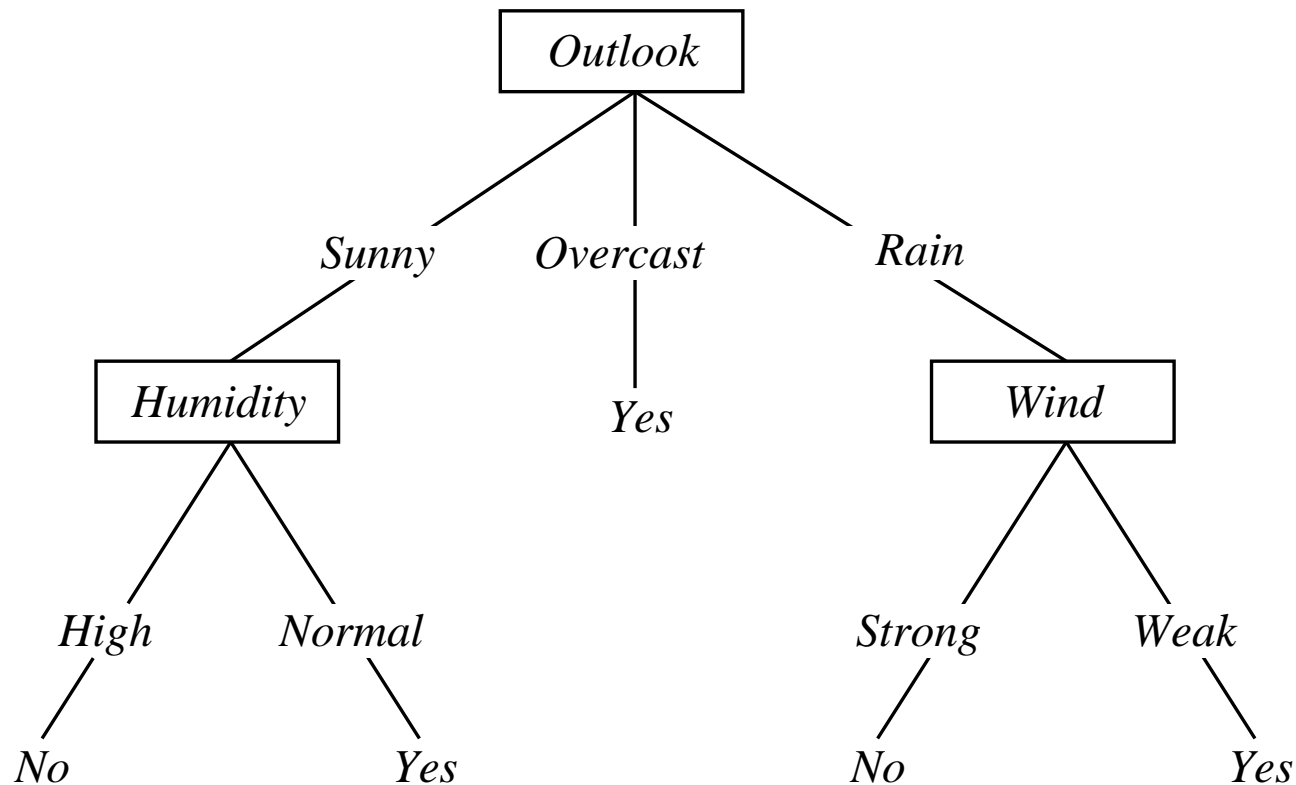
Rozwiązanie: można zastosować normalizację miary *Gain*

$$SplitInformation(S, a) = -\sum_{v \in Values(a)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

i użyć miary *GainRatio* zamiast *Gain*:

$$GainRatio(S, a) = \frac{Gain(S, a)}{SplitInformation(S, a)}$$

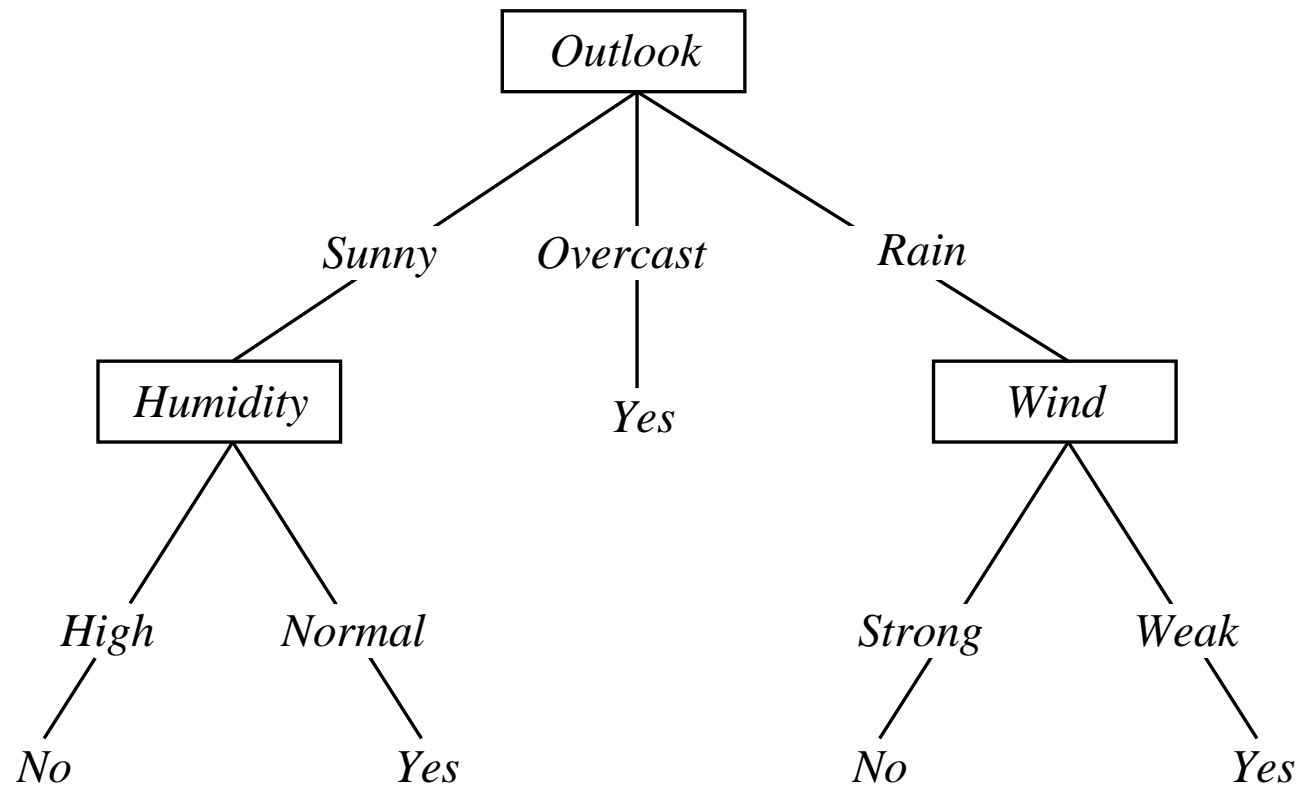
# Drzewa decyzyjne: klasyfikacja obiektu



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
Today	Sunny	Hot	Normal	Weak	??

Odpowiedź??

# Drzewa decyzyjne: klasyfikacja obiektu



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
Today	Sunny	Hot	Normal	Weak	??

Odpowiedź??

$Outlook = Sunny \rightarrow Humidity = Normal \rightarrow PlayTennis = Yes$

## Drzewa decyzyjne: wartosci brakujace

### Problem:

Co robić, kiedy informacja o klasyfikowanym obiekcie jest niepełna, np.:

<i>Day</i>	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Wind</i>	<i>PlayTennis</i>
<i>Today</i>	<i>?</i>	<i>Hot</i>	<i>High</i>	<i>Weak</i>	<i>??</i>



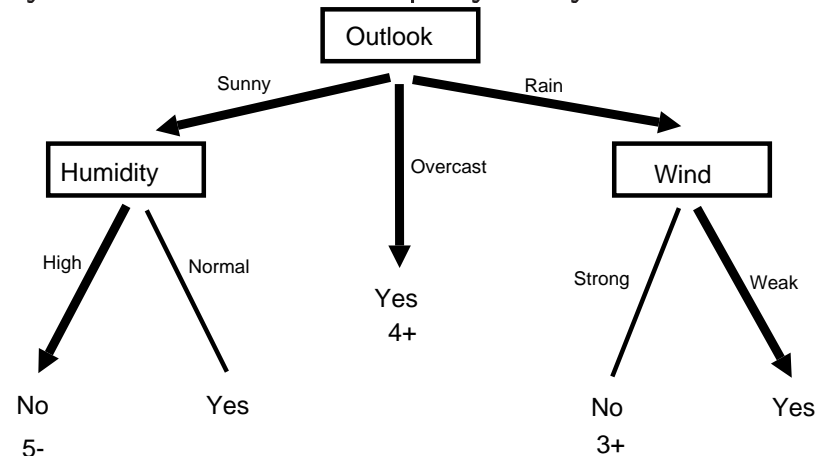
# Drzewa decyzyjne: wartosci brakujace

## Problem:

Co robić, kiedy informacja o klasyfikowanym obiekcie jest niepełna, np.:

<i>Day</i>	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Wind</i>	<i>PlayTennis</i>
<i>Today</i>	<i>?</i>	<i>Hot</i>	<i>High</i>	<i>Weak</i>	<i>??</i>

Pomysł: zejście wszystkimi ścieżkami przy atrybutach z nieustaloną wartością



Odpowiedź:??

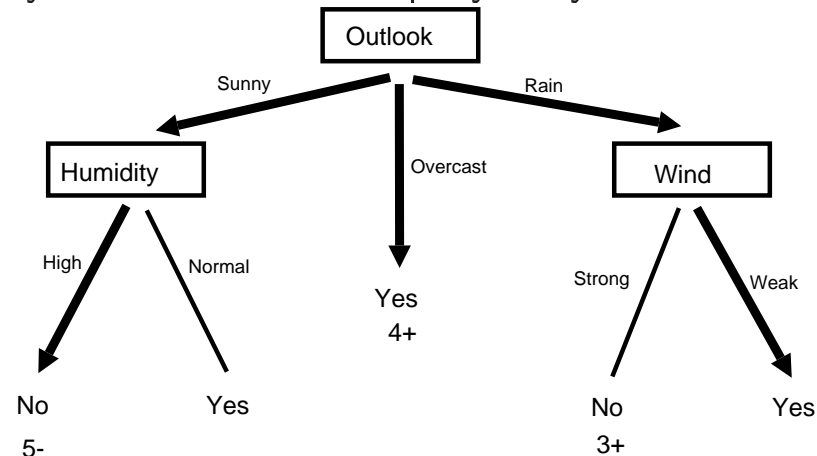
## Drzewa decyzyjne: wartosci brakujace

### Problem:

Co robić, kiedy informacja o klasyfikowanym obiekcie jest niepełna, np.:

<i>Day</i>	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Wind</i>	<i>PlayTennis</i>
<i>Today</i>	<i>?</i>	<i>Hot</i>	<i>High</i>	<i>Weak</i>	<i>??</i>

Pomysł: zejście wszystkimi ścieżkami przy atrybutach z nieustaloną wartością



### Odpowiedź:??

Maksym. z sumy rozkładów decyzji obiektów uczących w uzyskanych liściach:

$$5 \times No \wedge (4 + 3) \times Yes \rightarrow PlayTennis = Yes$$

# Drzewo decyzyjne: przycinanie

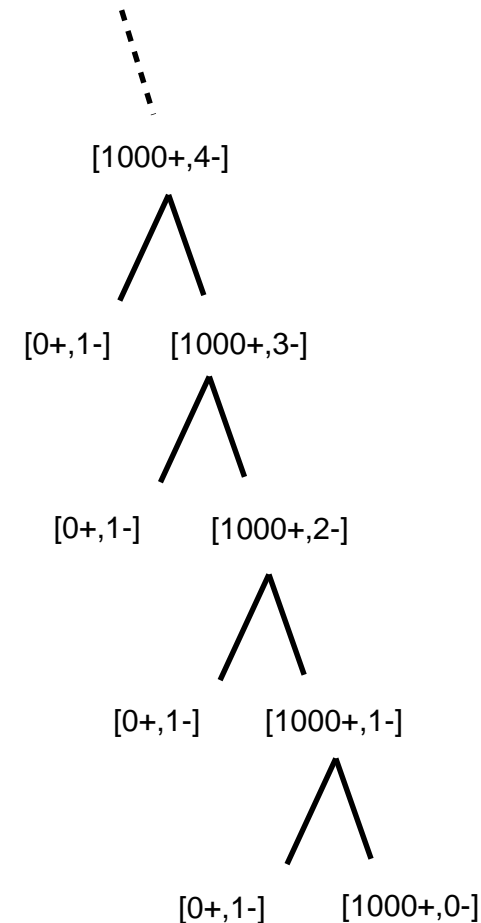
## Problem:

Bardzo rzadkie wyjątki lub błędy  
w przykładach uczących  
mogą spowodować niepotrzebne  
rozwiniecie gałęzi drzewa

## Pomysł:

Dodanie fazy walidacji drzewa decyzyjnego  
do procesu uczenia.

Węzły rozdzielające, które nie potwierdzą  
swojej przydatności w fazie walidacji,  
są zamieniane na liście.



## Drzewo decyzyjne: algorytm z przycinaniem

**function** DTL-PRUNED(*examples, attributes, default*) **returns** a decision tree

*building, validating*  $\leftarrow$  split *examples* into building and validating sets

*tree*  $\leftarrow$  DTL(*building, attributes, default*)

*tree*  $\leftarrow$  PRUNE-TREE(*validating, tree*)

**return** *tree*

---

**function** PRUNE-TREE(*validationset, tree*) **returns** a pruned decision tree

*leafparents*  $\leftarrow$  inner nodes in *tree* such that all their successors are leaf nodes

**while** *leafparents*  $\neq \{\}$

*cand*  $\leftarrow$  a node from *leafparents*

*d<sub>cand</sub>*  $\leftarrow$  the most frequent decision among the building objects assigned to *cand*

**if** assigning *d<sub>cand</sub>* to the objects going through the node *cand*

        does not worsen classification accuracy in *validationset* **then**

**replace** the subtree rooted at *cand* by the leaf node with the decision *d<sub>cand</sub>*

**if** all brother nodes of *cand* are leafs **then**

**add** the parent of *cand* to *leafparents*

**return** the modified *tree*