Systemy uczące się
wykład 2
Drzewa klasyfikacyjne - uzupełnienie

Jerzy Stefanowski
Instytut Informatyki PP
2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa
POPC.03.02.00-00-0001/20

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

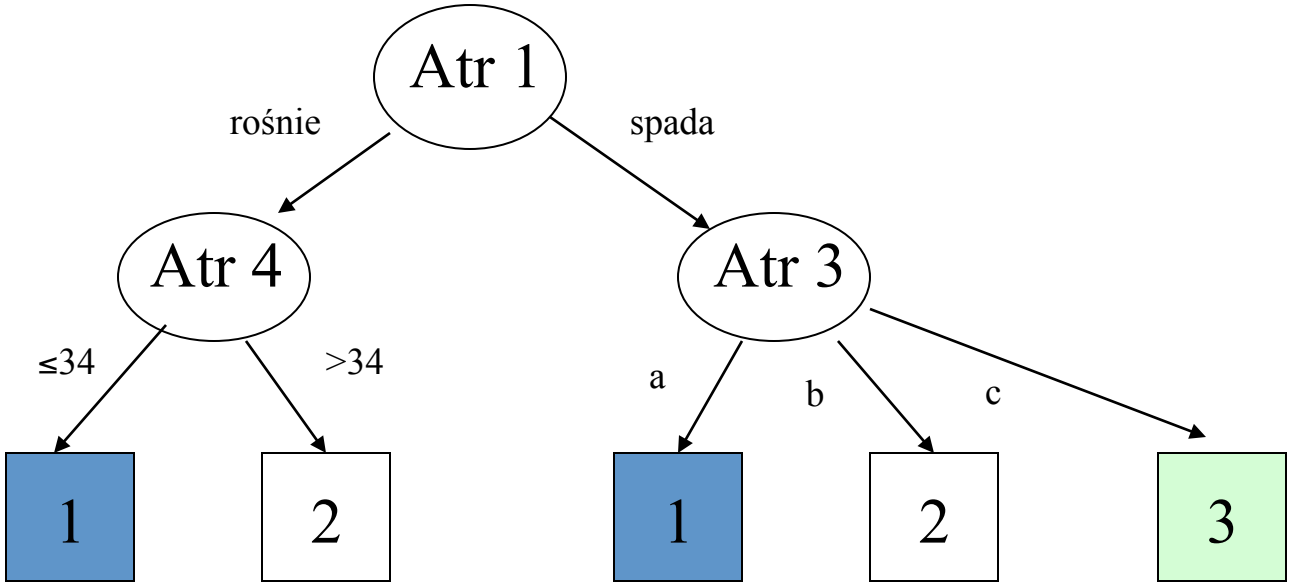
Plan wykładu

1. Drzewa decyzyjne
2. Algorytm ID3, entropia informacji
   - Uwzględnianie danych niedoskonałych
   - Dyskretyzacja atrybutów liczbowych
3. Inne rozszerzenia → C4.5
4. Przeuczenie klasyfikatorów i tzw. upraszanie budowy drzew
5. Podsumowanie

Co to jest drzewo decyzyjne?

Jest to struktura grafu skierowanego z góry na dół:

- Węzły reprezentują pytanie o wartości cech
- Z węzłów wychodzą gałęzie które reprezentują wynik pytania
- Liście reprezentują klasy decyzyjne



Drzewa - zagadanienia

W miarę dojrzała metodologia, wiele implementacji, liczne zastosowania

Podstawowe problemy:

Jak je tworzyć automatycznie?

- Algorytmy
- Kryterium wyboru w węźle

Przeuczenie (dobra ilustracja – wielkość drzewa)

- Tzw. redukcja drzewa (ang. pruning)

Metody indukcji drzew decyzyjnych

- Podejście obejmuje dwa etapy:
  - **Konstrukcja drzewa (rekurencyjna procedura)**
    - Na początku wszystkie przykłady w węźle.
    - Rekurencyjnie dziel przykłady w oparciu o wybrane testy na wartościach atrybutu (kryterium wyboru najlepszego atrybutu).
    - Zatrzymaj gdy wszystkie przykłady „w gałęzi” należą do jednej klasy
  - Upraszczanie drzewa - „Tree pruning”
    - Usuwanie poddrzew, które mogą prowadzić do błędnych decyzji podczas klasyfikacji przypadków testowych.
  - Przykłady algorytmów: ID3, C4.5, CART,…

Przykład budowy DT – Quinlan „ play golf”

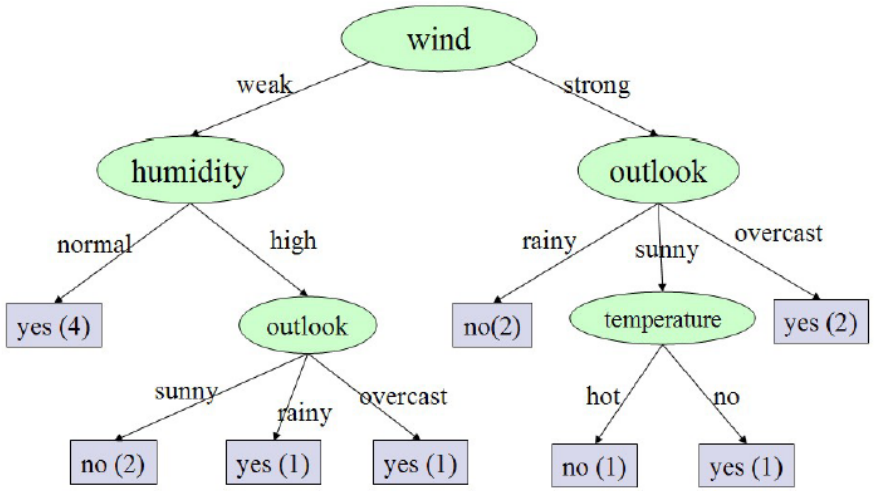
Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

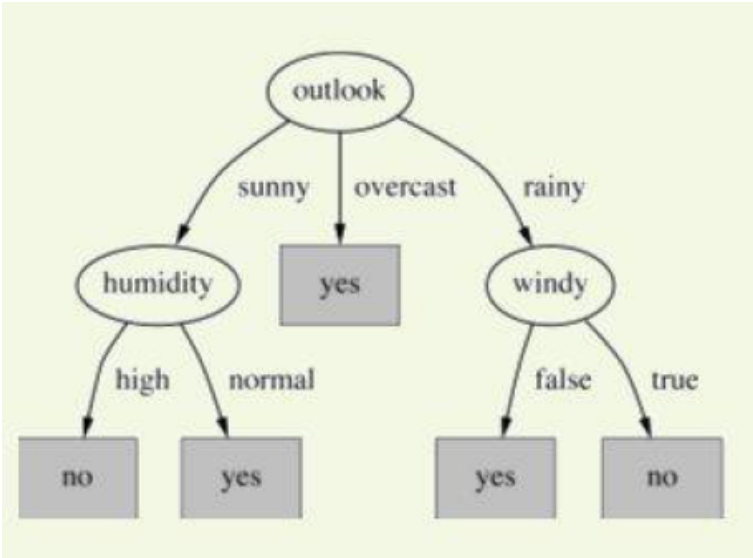
Uproszczona Tabela danych

Poszukiwanie dobrych drzew

Play or not (Quinlan)

x	outlook	Temperature	humidity	wind	play(x)
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	cold	normal	weak	yes
6	rain	cold	normal	strong	no
7	overcast	cold	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cold	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





Ogólny schemat

TDIDT - Top Down Iterative Decision Tree

function DT(*E*: zbiór przykładów) **returns** drzewo;
    T' := buduj_drzewo(*E*);
    T := obetnij_drzewo(*T'*);
    return *T*;

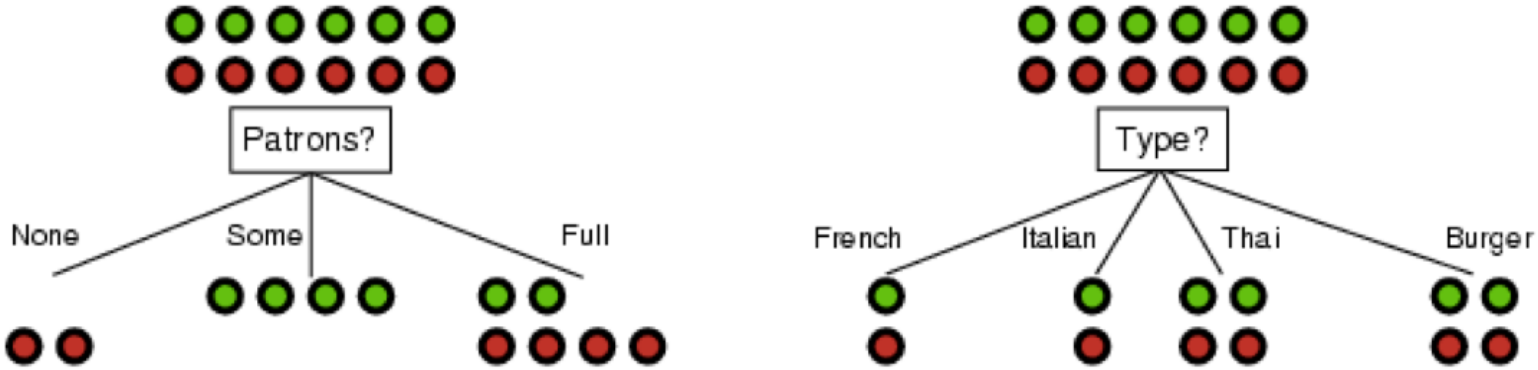
function buduj_drzewo(*E*: zbiór przyk.) **returns** drzewo;
        T := generuj_tests_atr_A(*E*);
        t := najlepszy_test(*T*, *E*);
        P := podział *E* indukowany przez *t*;
        if kryterium_stopu(*E*, *P*)
        then return liść(info(*E*))
        else
            for all E_j **in** P: t_j := buduj_drzewo(E_j);
            return węzeł(*t*, {(*j*,t_j)};

Intuicja wyboru atrybutu

Przykład decyzji o wyborze restauracji [Rusell, Norvig]

Split condition -

Dobry atrybut powinien podzielić zbiór przykładów S na podzbiory S1, S2,…, które są możliwie jednoznaczne (purity) wskazać klasy decyzyjne – poszukiwanie możliwie najprostszego drzewa zgodnego z przykładami uczącymi



Which split is more informative: *Patrons?* or *Type?*

Entropia (C. Shannon)

- Entropia (zawartość informacyjna, *information content*): miara oceniająca zbiór przykładów pod kątem ‘czystości’ (jednolitości przynależności do klas decyzyjnych)
- Dla dwóch klas decyzyjnych (pozytywna, negatywna) – *p* liczba przykładów pozytywnych, *n* - negatywnych:

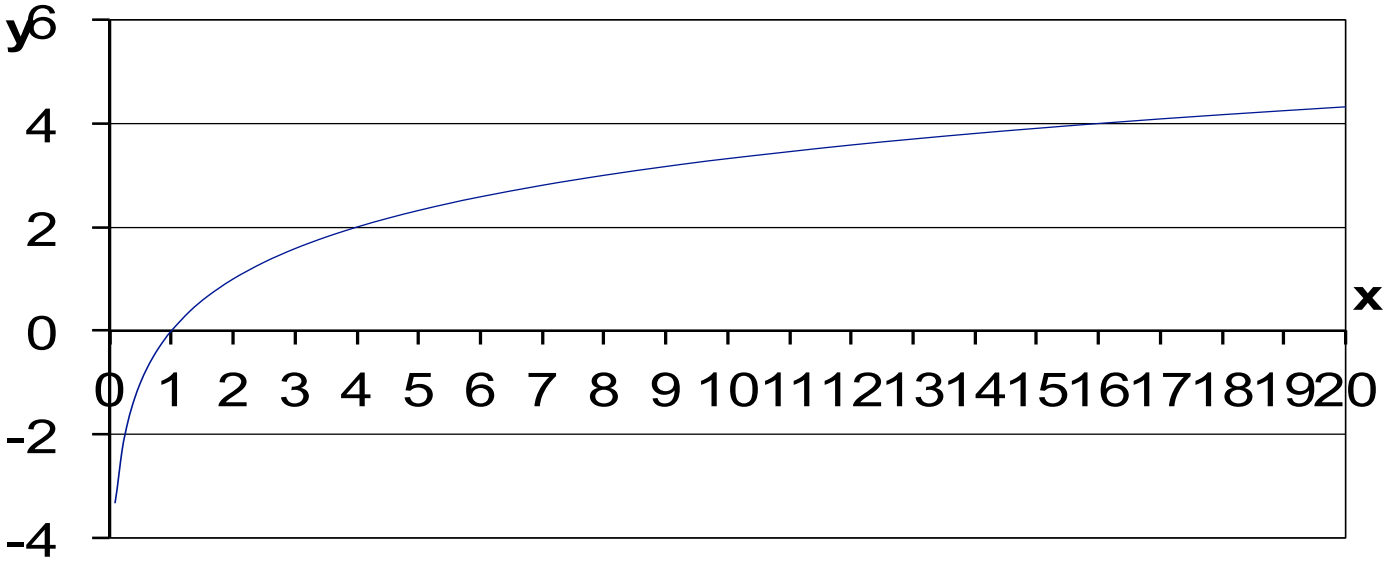
$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Wersja ogólna wieloklasowa p_i – prawdopodobieństwo, że przykład należy do i-tej klasy:

$$I = -\sum_{i=1}^{K} p_i \cdot \left(\log_2 p_i\right)$$

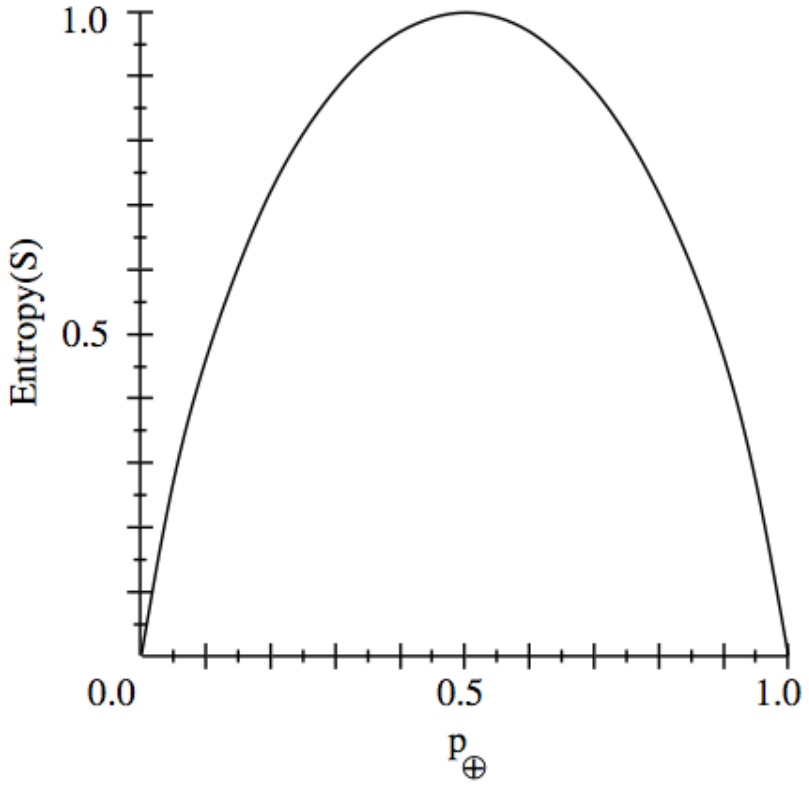
Przypomnienie logarytmów

- Funkcja log. $y = \log_a x$
- a – podstawa logarytmu $x = a^y$
- Rozważmy funkcję logarytmiczną dla $a = 2$ (tj. log2x)



x	1/8	1/4	1/2	1	2	4	8
y	-3	-2	-1	0	1	2	3

Entropia – interpretacja i własności mat.



Analiza binarnej entropii dla dwóch klas

Entropia dla przykładu golf

Nie oceniamy podziału atrybutem, tylko rozkład wartości klas decyzyjnych

Dwie klasy : *yes* and *no*

Z 14 przykładów 9 etykietowanych jako *yes*, reszta jako *no*

$$p_{yes} = -\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) = 0.41$$

$$p_{no} = -\left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) = 0.53$$

$$E(S) = p_{yes} + p_{no} = 0.94$$

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

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

Zysk informacji - Information gain

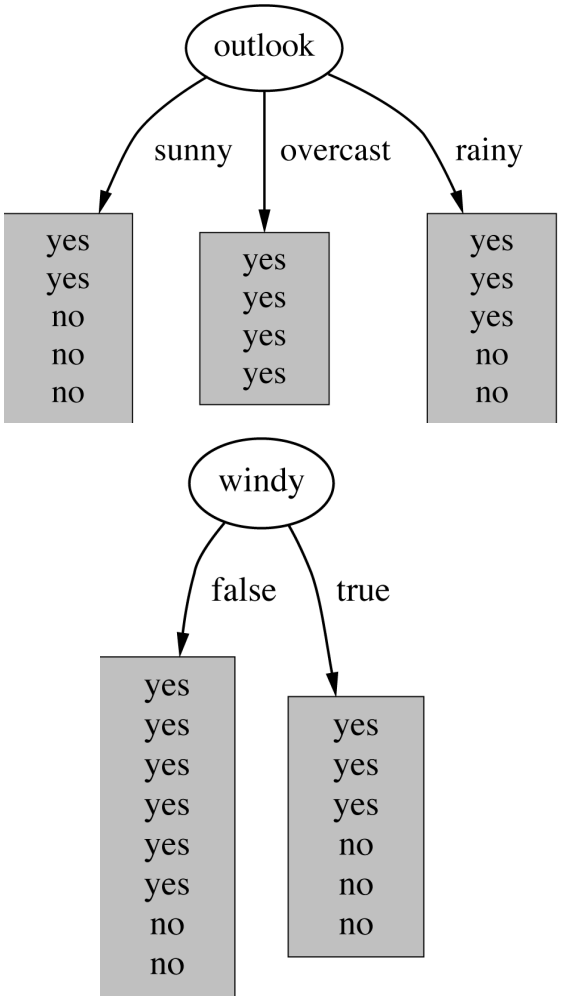
- Entropia warunkowa: entropia po podziale zbioru przykładów przy pomocy atrybutu A (załóżmy że A przyjmuje *v* możliwych wartości):

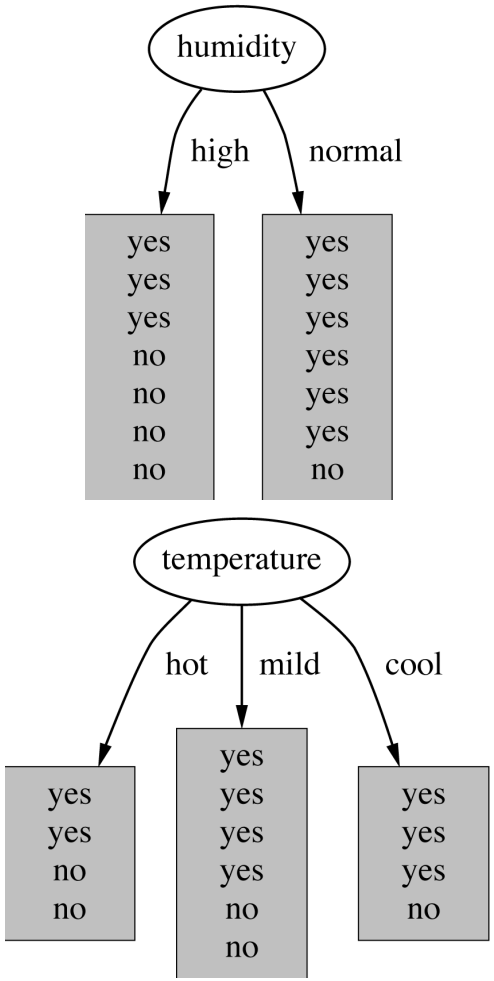
$$EntropiaWarunkowa(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Zysk informacyjny (*Information Gain*): redukcja entropii przy wykorzystaniu danego atrybutu:

$$IG(A) = I - EntropiaWarunkowa(A)$$

Który atrybut należy wybrać?





Przykład oceny atrybutu "Outlook"

- "Outlook" = "Sunny":

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971$$

- "Outlook" = "Overcast":

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0$$

Uwaga: log(0) jest nieskończone lecz 0*log(0) dąży do zera

- "Outlook" = "Rainy":

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971$$

- Entropia warunkowa dla podziału wartościami atrybutu

$$\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971$$
$$= 0.693$$

Obliczanie zysku informacyjnego miary entropii

- Zysk informacji -> Information gain:

(information before split) – (information after split)

$$\text{gain}(\text{"Outlook"}) = \text{info}([9,5]) - \text{info}([2,3],[4,0],[3,2]) = 0.940 - 0.693 = 0.247$$

- Ostateczne wartości zysku

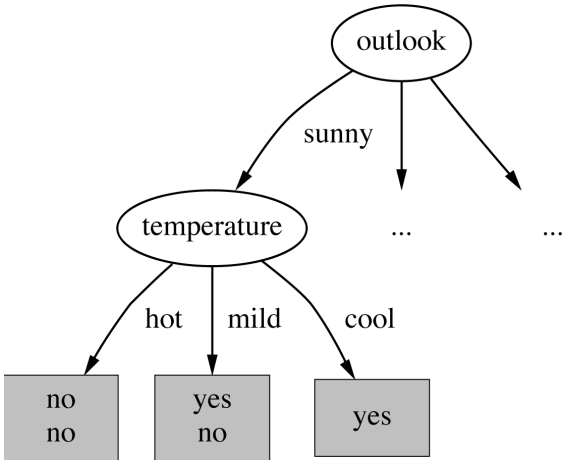
  Gain(„Temperature”)=0.029

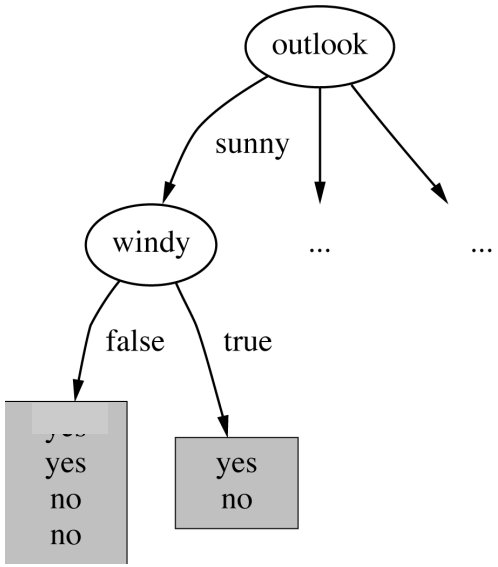
  Gain(„Humidity”)=0.152

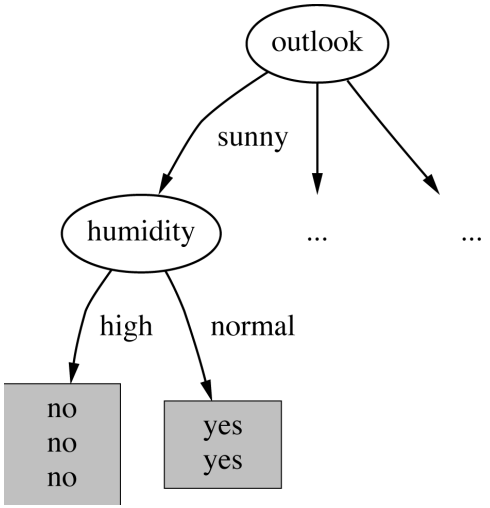
  Gain(„Windy”)=0.048

  – Co wybieramy?

Rozbuduj drzewo





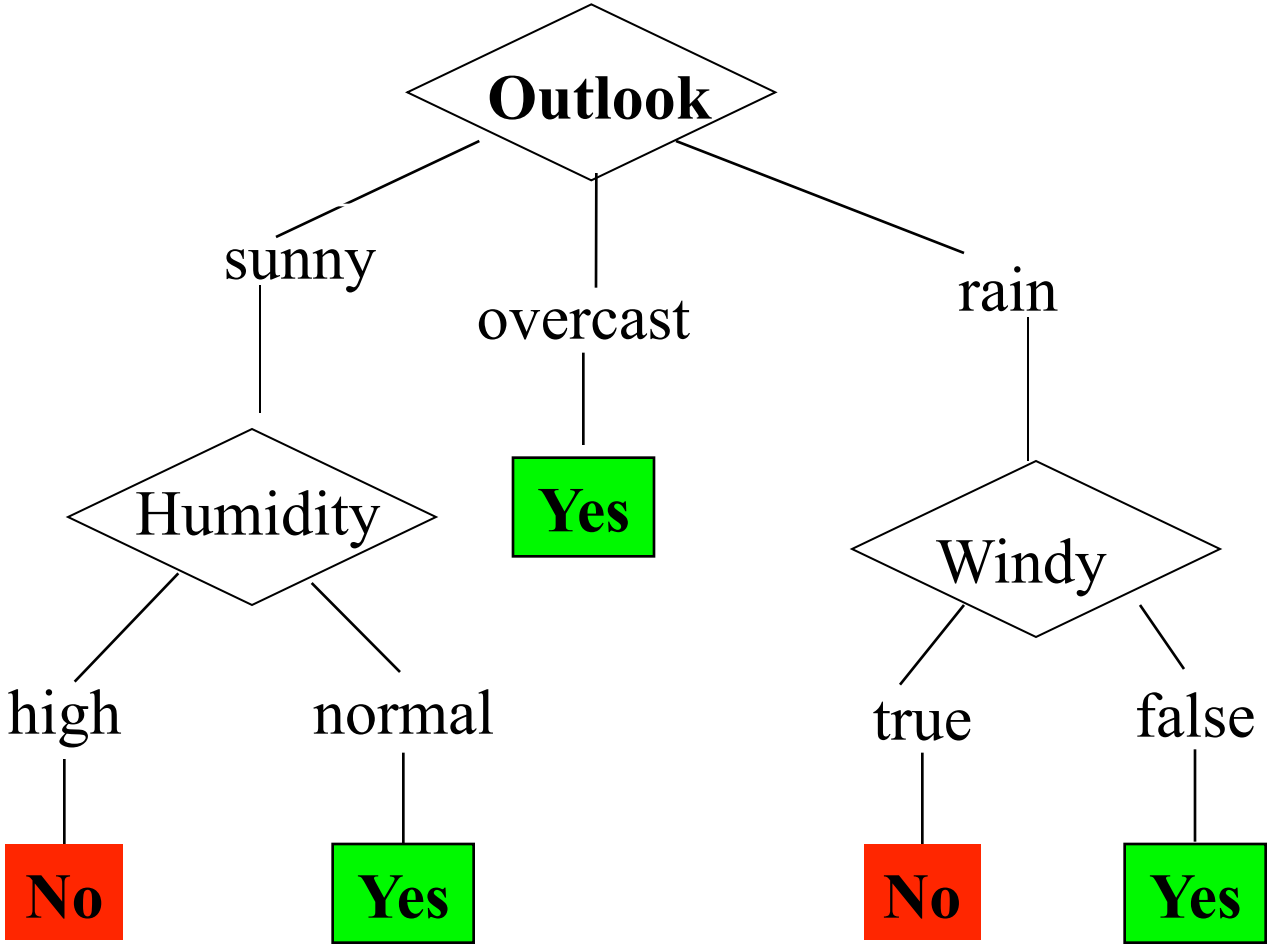


gain("Humidity") = 0.971

gain("Temperature") = 0.571

gain("Windy") = 0.020

Ostateczne drzewo

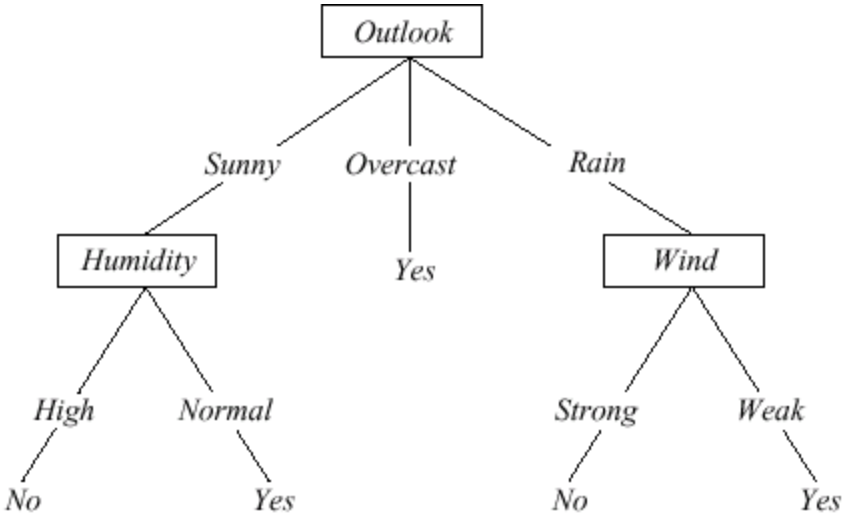


Wykorzystanie drzewa

- Bezpośrednio:
  - sprawdzaj wartości atrybutu nowego przykładu zaczynając od korzenia do liści
- Pośrednio:
  - zamień strukturę drzewa na zbiór reguł decyzyjnych (upraszczając nadmiarowe warunki)
  - reguły uważa się za czytelniejszą reprezentacje

DT => reguły

Zamień DT na reguły i uprość: łatwo ocenić, które reguły można usunąć i optymalizować pozostałe.



IF $(Outlook = Sunny) \wedge (Humidity = High)$ THEN $PlayTennis = No$

IF $(Outlook = Sunny) \wedge (Humidity = Normal)$ THEN $PlayTennis = Yes$

Inne kryteria podziału

Indeks Gini-ego (CART, alg. dla tzw. datamining)

$$Gini = 1 - \sum_{i=1}^{K} p_i^2$$

Także warunkowa postać po wyborze podziału A

Silnie wielowartościowe atrybuty

- Problematyczne : atrybuty o relatywnie większej dziedzinie niż inne
- Podzbiory mało liczne po podziale mogą być „czystsze”
  - ⇒Preferencja miary entropi / zysku informacyjnego
  - ⇒Słabe własności generalizujące
- Wykorzystanie miary gain ratio lub binaryzacja drzewa
- Gain-ratio (Quinlan)

$$GainRatio(S,A) = \frac{Gain(S,A)}{IntrinsicInfo(S,A)}.$$

$$IntrinsicInfo(S,A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

Binary Tree – budowa drzew binarnych

- Drzewa binarne mogę być skuteczniejsze w klasyfikacji nowych faktów
- Podział binarny w węźle drzewa:
  - Atrybuty liczbowe A, reprezentacja w postaci value(A)<x gdzie x jest wartością z dziedziny A.
  - Atrybuty nieliczbowe A, warunek w postaci value(A)$\in$X gdzie X$\subset$domain(A)

Binary tree (Quinlan's C4.5 output)

```
Pruned decision tree:

A9 = t:
|   A15 > 228 : + (106.0/3.8)
|   A15 <= 228 :
|   |   A14 <= 102 :
|   |   |   A4 in {l,t}: + (0.0)
|   |   |   A4 = u:
|   |   |   |   A6 in {c,d,cc,i,k,m,q,w,x,e,aa}: + (46.4/3.1)
|   |   |   |   A6 in {j,ff}: - (2.0/1.0)
|   |   |   |   A6 = r: + (0.0)
|   |   |   A4 = y:
|   |   |   |   A6 in {c,i,aa,ff}: - (7.0/3.4)
|   |   |   |   A6 in {d,j,w,x}: + (4.0/1.2)
|   |   |   |   A6 in {cc,k,m,r,q,e}: + (0.0)
|   |   A14 > 102 :
|   |   |   A6 in {j,r}: + (0.0)
|   |   |   A6 in {c,d,k,m,e,aa,ff}:
|   |   |   |   A14 <= 132 : - (4.1/1.2)
|   |   |   |   A14 > 132 :
|   |   |   |   |   A3 <= 1.625 :
|   |   |   |   |   |   A14 <= 292 : - (13.0/1.3)
|   |   |   |   |   |   A14 > 292 :
|   |   |   |   |   |   |   A13 = g: + (2.0/1.0)
|   |   |   |   |   |   |   A13 = s: - (6.0/2.3)
|   |   |   |   |   |   |   A13 = p: - (0.0)
|   |   |   |   |   A3 > 1.625 :
|   |   |   |   |   |   A6 in {k,m}: + (5.0/1.2)
|   |   |   |   |   |   A6 = ff: + (0.0)
|   |   |   |   |   |   A6 in {c,d,e,aa}:
|   |   |   |   |   |   |   A2 <= 32.08 : + (9.5/4.1)
|   |   |   |   |   |   |   A2 > 32.08 : - (8.0/3.5)
|   |   |   A6 in {cc,i,q,w,x}:
|   |   |   |   A8 <= 10.75 : + (36.0/9.3)
|   |   |   |   A8 > 10.75 : - (2.0/1.0)
A9 = f:
|   A4 in {u,y}: - (237.0/17.3)
|   A4 = l: + (2.0/1.0)
|   A4 = t: - (0.0)
```

- Crx (Credit Data) UCI ML Repository
- =źródło własne

Binaryzacja atrybutu ilościowego

- Punkt podziału - Split dla atr. temperature :

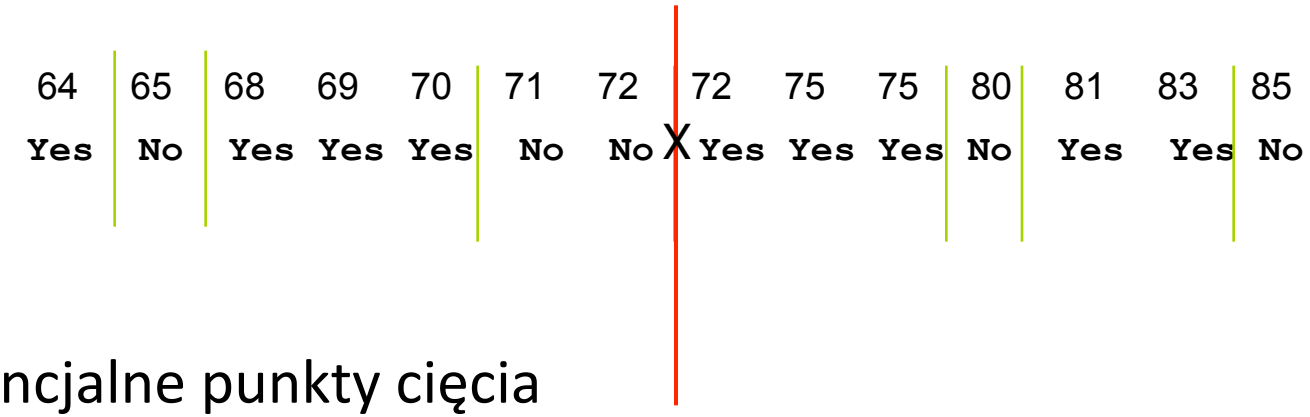
64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	**No**	**Yes**	**Yes**	**Yes**	**No**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**No**

  - Np. temperature < 71.5: yes/4, no/2
    temperature ≥ 71.5: yes/5, no/3

  - Info([4,2],[5,3])
    = 6/14 info([4,2]) + 8/14 info([5,3])
    = 0.939
- Wstaw próg między istniejące przykłady
- Efektywne obliczeniowo

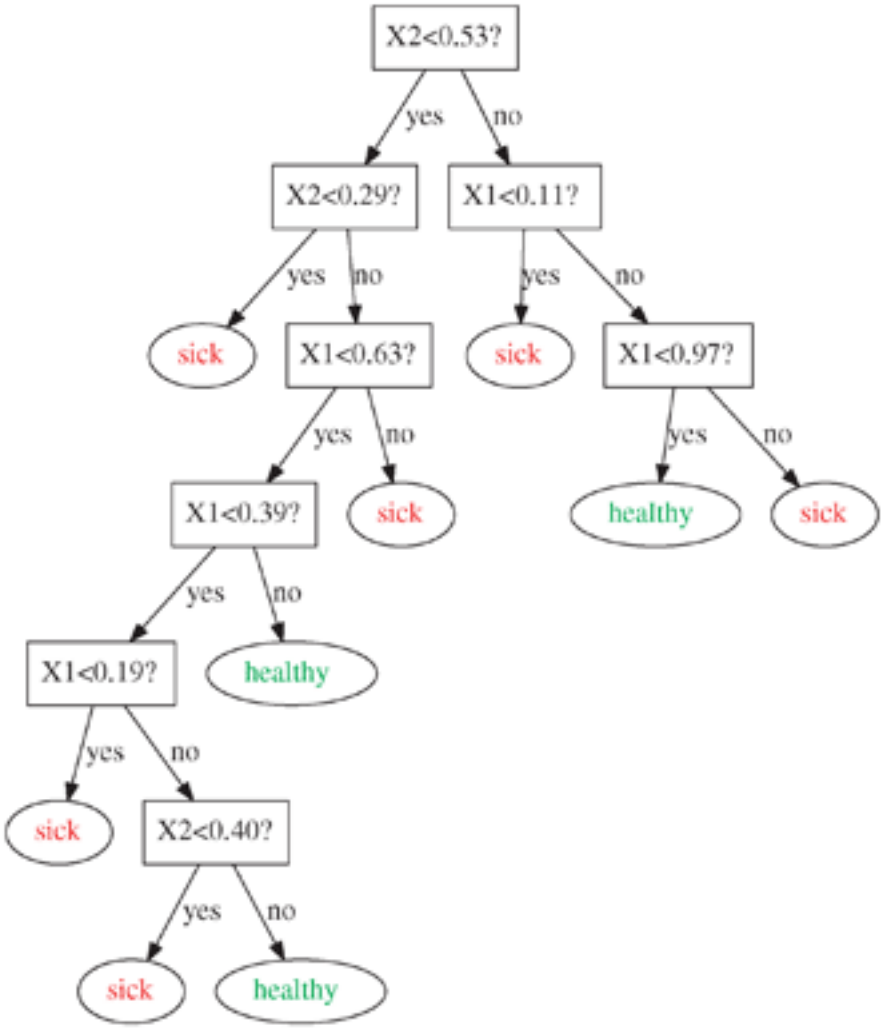
Szybsze obliczenia

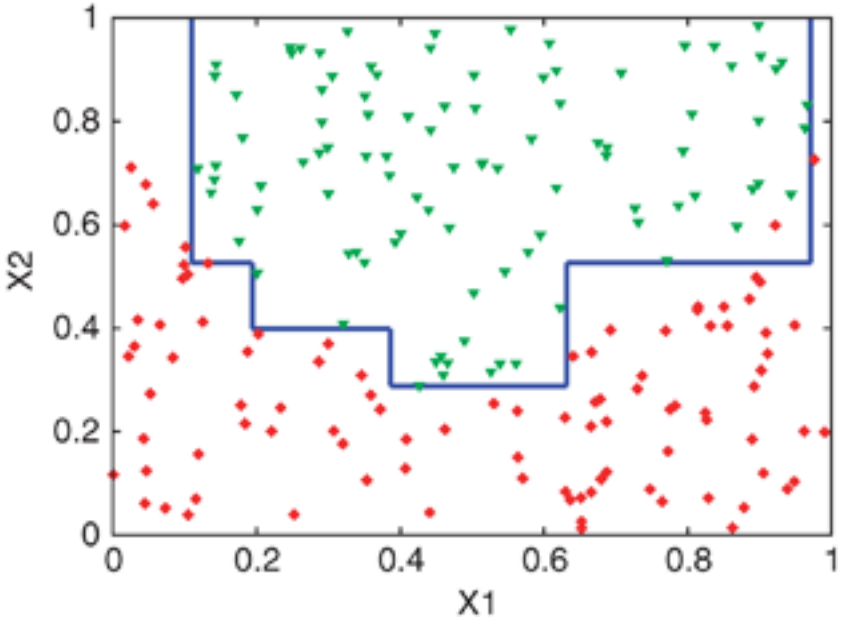
- Własności mat. entropii(Fayyad & Irani, 1992)



Potencjalne punkty cięcia

Przykład medyczny





Inne wyzwanie …

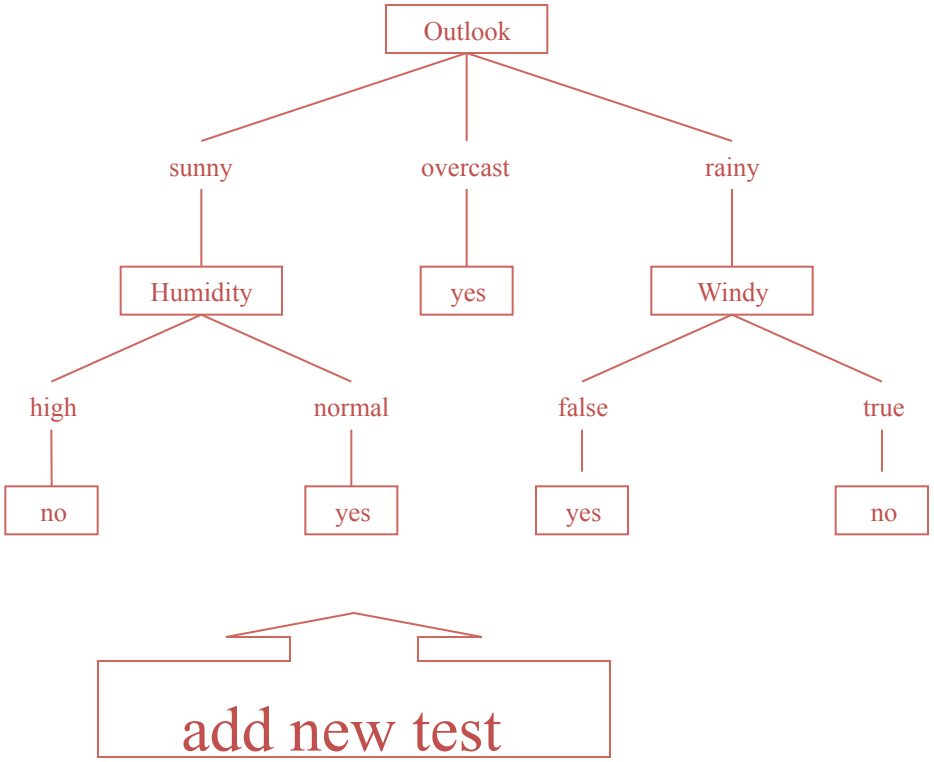
Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	**cool**	**normal**	**false**	**Yes**
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Co będzie dla sprzecznych przykładów

Sprzeczne opisy przykładów

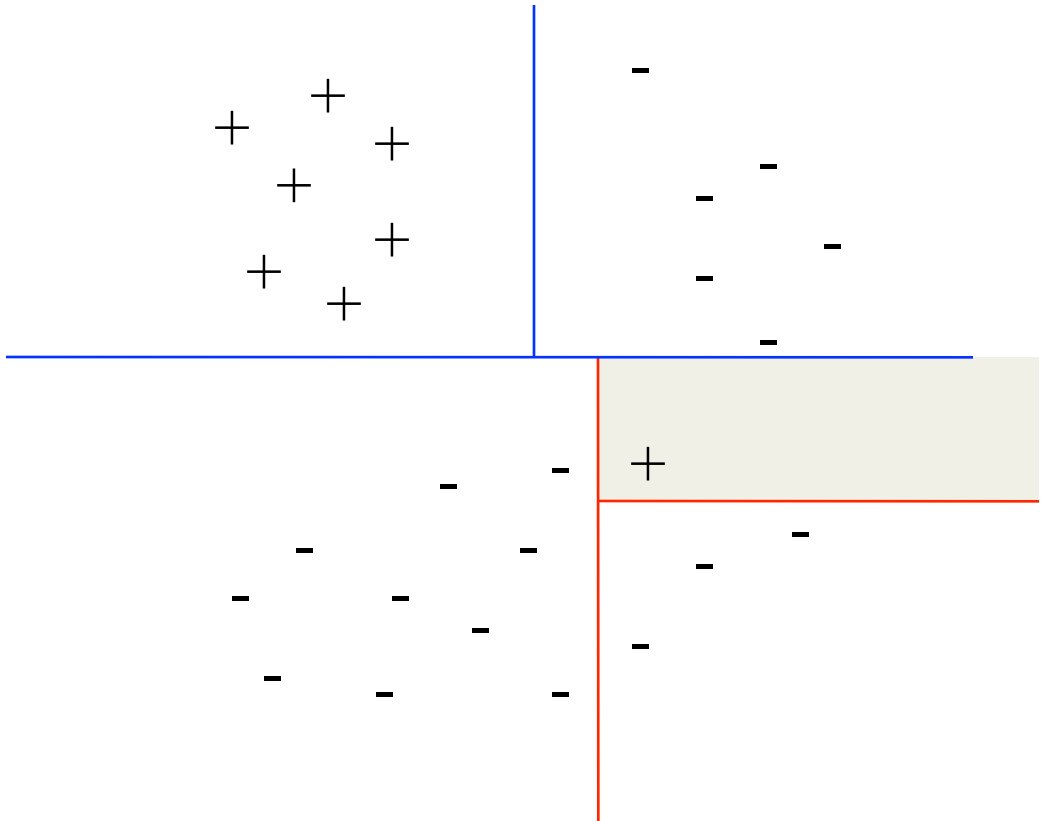
Nowy sprzeczny przykład:

Outlook = *Sunny;* Temperature = *Cool;* Humidity = *Normal;* Wind = *False;* PlayTennis = *No*



Inne przyczyny przeuczenia drzewa

- ***Nietypowe przykłady*** – prowadzą do rozbudowanych poddrzew z małą liczbą przykładów wspierające liście



Przeuczenie klasyfikatora (Overfitting)

- Dobry klasyfikator (drzewo) musi nie tylko wystarczająco spójnie odwzorowywać dane uczące, lecz także trafnie klasyfikować nowe dane (niewidziane w trakcie procesu uczenia się).
- Innymi słowami – klasyfikator musi mieć niski błąd uczący, lecz przede wszystkim niski błąd uogólnienia na nowe dane (testowe)
  - Błąd treningowy vs. błąd testowy
- Nadmiernie rozbudowane drzewo, dopasowane do trudnych przykładów uczących traci zdolności uogólniania.

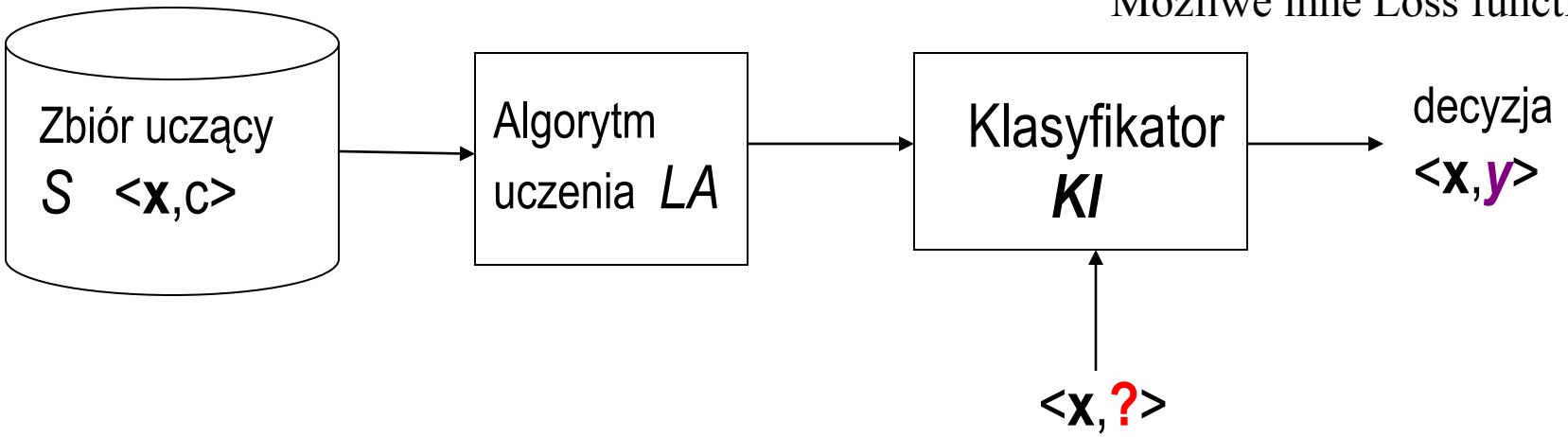
Predykcja nowych faktów - klasyfikatory

Predykcja klasyfikacji nowych obiektów (zbiór testowy)
→ *Cross validation*

Miara oceny, np:
trafność klasyfikowania

$$\eta = \frac{N_c}{N_t}$$

Możliwe inne Loss functions



Przykłady $S = \{\langle \mathbf{x}_1, c_1 \rangle, \langle \mathbf{x}_2, c_2 \rangle, \cdots, \langle \mathbf{x}_n, c_n \rangle\}$
$\mathbf{x}_i = <x_{i1}, x_{i2}, \ldots, x_{im}>$ opisywane przez m atrybutów
Atrybuty różnego typu
c_i – etykieta jednej z klas $\{C_1, \ldots, C_K\}$

Nowe instancje → $Kl(\mathbf{x})$

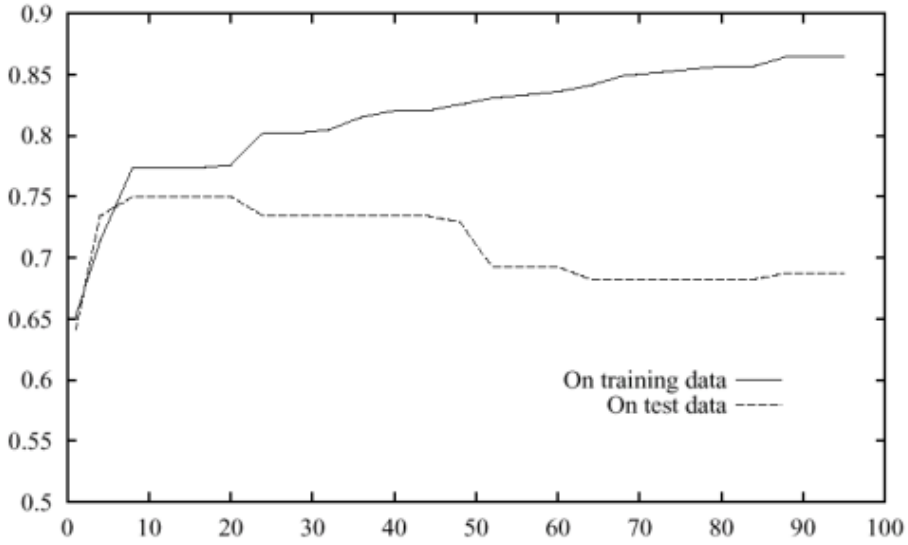
albo

Przykłady testowe
→ $<\mathbf{x}_j, ?>$ / + znamy poprawną etykietę klasy C_{ji}

$Kl(\mathbf{x})$? C_{ji} [Loss function]

Overfitting the Data – nadmierne dopasowanie do danych uczących

- Podstawowy algorytm ID3 → Rozbudowuj gałąź drzewa do pełnego rozróżnienia przykładów
  - Sensowe na spójnych przykładów i celów dokładnego opisu
- Rzeczywiste dane (niespójne, szum informacyjny) oraz cel klasyfikowania przykładów
  - Drzewa mają tendencje do przeuczenia / nadmiernego dopasowania do specyficznych przykładów *overfit* the learning examples
  - Occam razor – zasada brzytwy Occama (z konkurujących drzew wybierz prostsze; ma lepsze własności generalizacyjne)



Brzytwa Ockhama

Czemu preferować prostsze drzewa?

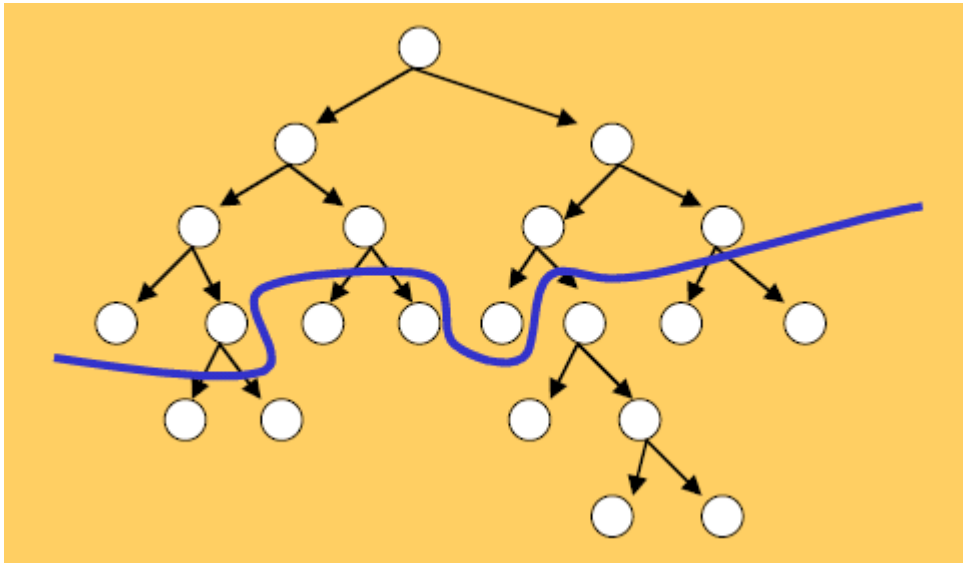
1. Mało prostych hipotez, więc mała szansa, że przypadkiem pasują do danych.
2. Proste drzewa nie powinny zbytnio dopasować się do danych.
3. Przetrenowanie modelu dla zbyt złożonych drzew, zła generalizacja.

Ale:

1. Dla małych zbiorów o wielu atrybutach można tworzyć wiele prostych opisów danych.

Tree pruning – upraszczanie drzewa

- Mechanizm „walki” z przeuczeniem
- Po uproszczeniu struktury drzewa może wzrosnąć trafność na przykładach testowych!



Unikanie przetrenowania

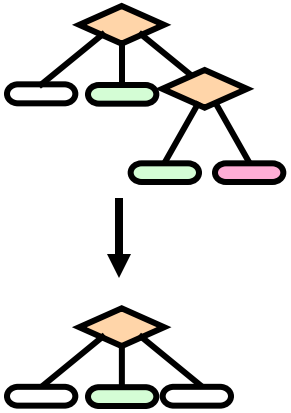
Jak uniknąć przetrenowania i radzić sobie z szumem?

1. Zakończ rozwijanie węzła jeśli jest zbyt mało danych by wiarygodnie dokonać podziału.
2. Zakończ jeśli czystość węzłów (dominacja jednej klasy) jest większa od zadanego progu – pre- pruning
   DT => drzewo prawd. klas.
3. Utwórz drzewo a potem je przytnij (post - pruning)

1. Przycinaj korzystając z wyników dla k-cv lub dla zbioru walidacyjnego.
2. Korzystaj z MDL (Minimum Description Length):
   Min Rozmiar(Drzewa) + Rozmiar(Drzewa(Błędów))
3. Oceniaj podziały zaglądając poziom (lub więcej) w głąb.

Reduced-Error Pruning

- Post-Pruning, Cross-Validation Approach
- Split Data into Training and Validation Sets
- Function *Prune*(*T, node*)
  - Remove the subtree rooted at *node*
  - Make *node* a leaf (with majority label of associated examples)



- Algorithm *Reduced-Error-Pruning* (*D*)
  - Partition *D* into D_{train} (training / "growing"), $D_{validation}$ (validation / "pruning")
  - Build complete tree *T* using *ID3* on D_{train}
  - UNTIL accuracy on $D_{validation}$ decreases DO

    FOR each non-leaf node *candidate* in *T*

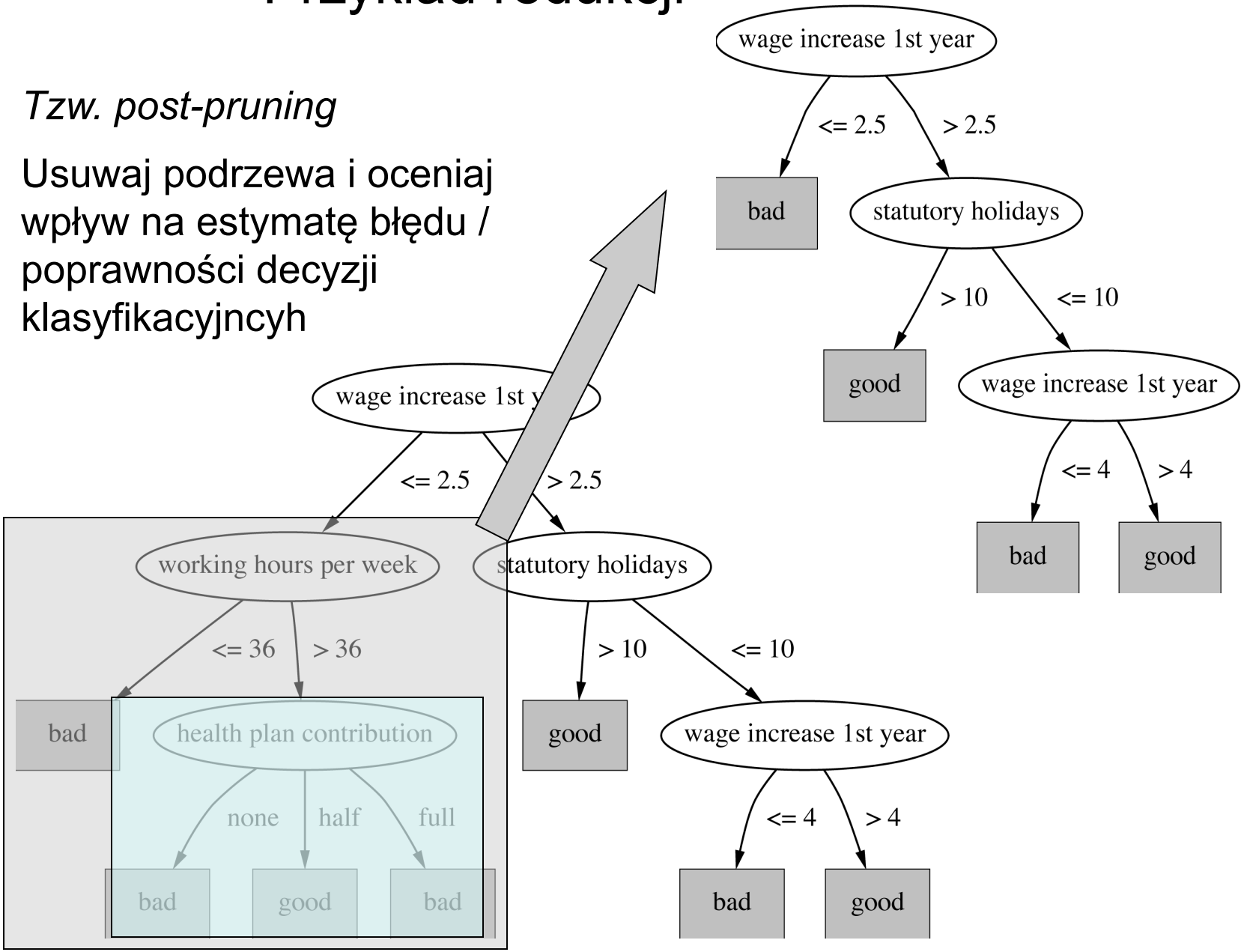
    ***Temp*[*candidate*] ← *Prune* (*T*, *candidate*)**

    ***Accuracy*[*candidate*] ← *Test* (*Temp*[*candidate*], $D_{validation}$)**

    $T \leftarrow T' \in Temp$ with best value of *Accuracy* (best increase; *greedy*)
  - RETURN (pruned) *T*

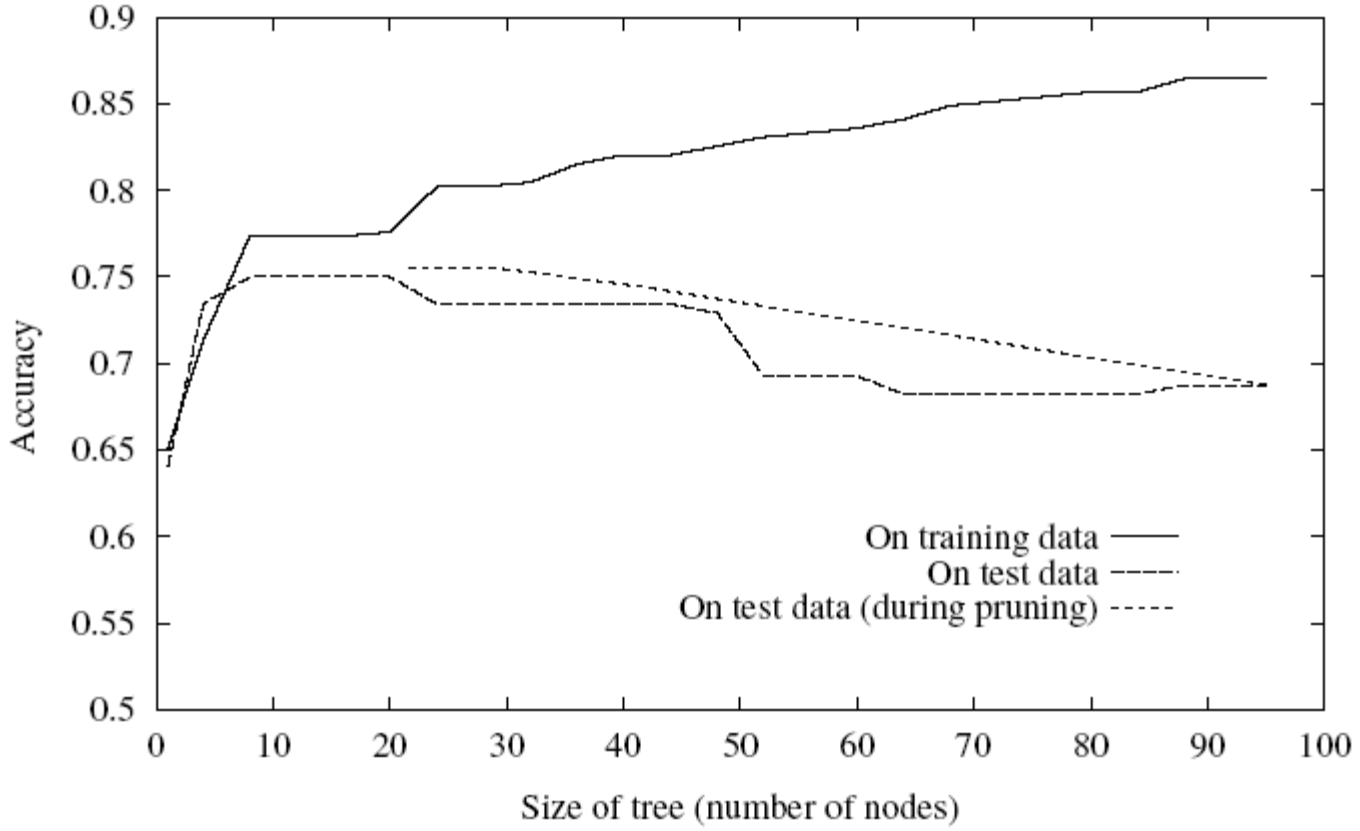
Przykład redukcji

- *Tzw. post-pruning*
- Usuwaj podrzewa i oceniaj wpływ na estymatę błędu / poprawności decyzji klasyfikacyjncyh

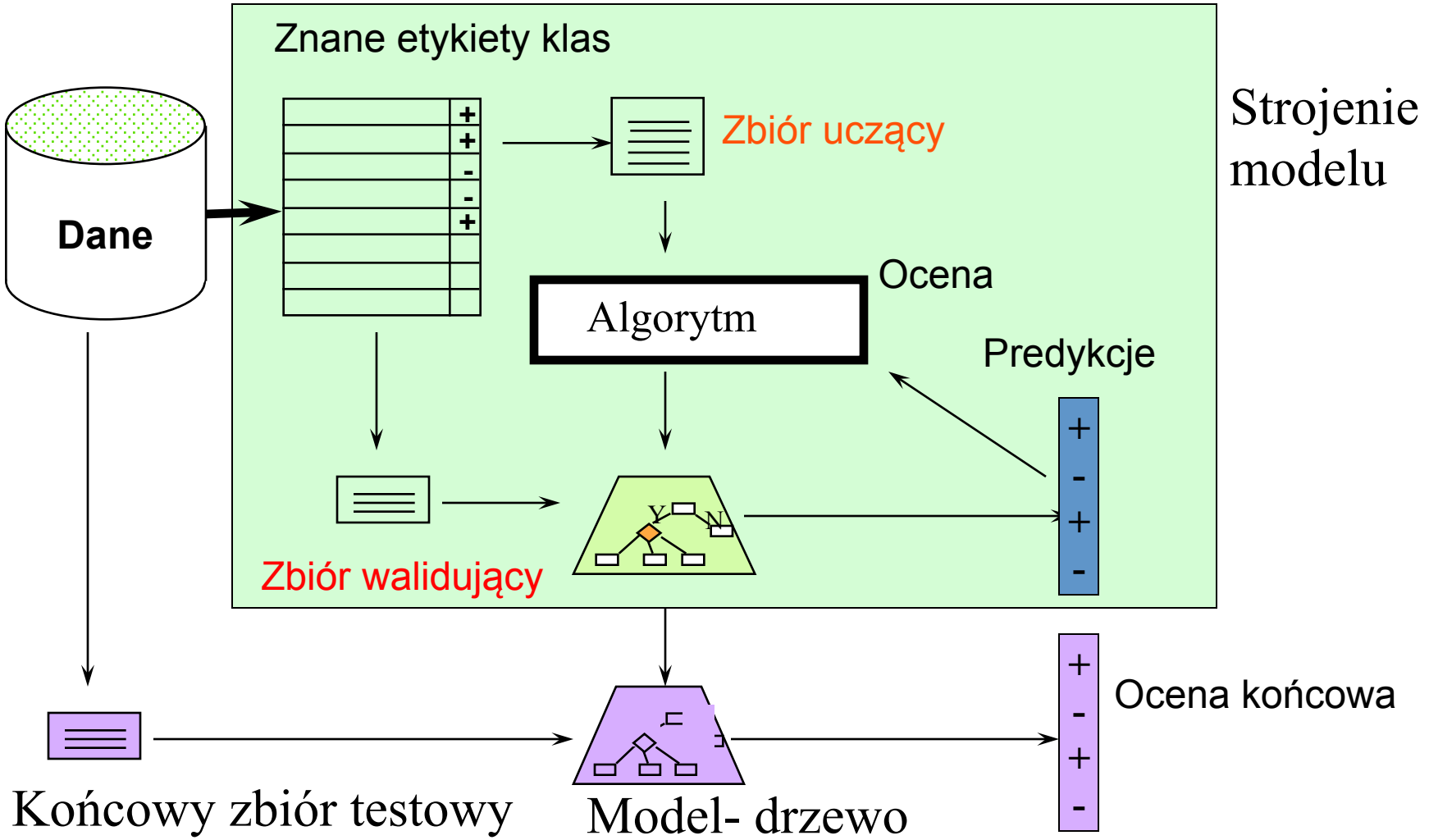


Reduced post-pruning

Zbiór przykładów testowych – nie można wykorzystywać w trakcie uczenia, potrzebny tzw. zbiór przykładów walidacyjnych



Trzy rodzaje danych: treningowe, walidacyjne, testowe

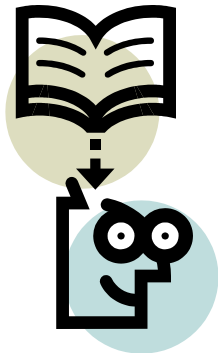


Przykłady zastosowań

- Wiele przykładów analizy podejmowania decyzji o diagnozowaniu chorób, także terapii, oraz farmacja i budowa związków - leków:
- Przykładowe omówienia:
  - I.Kononenko, I.Bratko, M.Kukar: Application of Machine Learning to Medical Diagnosis. w: Michalski R.S., Bratko I, Kubat M. (red.), Machine learning and data mining, John Wiley & Sons, 1998, s. 389-408.
  - Langley, P., Simon, H. A., Fielded applications of machine learning, w: Michalski R.S., Bratko I, Kubat M. (red.), Machine learning and data mining, John Wiley & Sons, 1998 , s. 113-129.
- Spójrz także na dodatkowe materiały – na podanym linku w ekursy

Trochę książek

- Uczenie maszynowe i sieci neuronowe. Krawiec K., Stefanowski J., Wydawnictwo Politechniki Poznańskiej, Poznań, 2003 (kolejne wydanie 2004)
- Systemy uczące się. Cichosz P., WNT, Warszawa, 2000
- Statystyczne systemy uczące się. Koronacki J., Ćwik J. WNT Warszawa 2008
-



Pytanie i komentarze?

Dalszy kontakt:

jerzy.stefanowski@cs.put.poznan.pl

http://www.cs.put.poznan.pl/jstefanowski/

Fundusze Europejskie Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska Europejski Fundusz Rozwoju Regionalnego