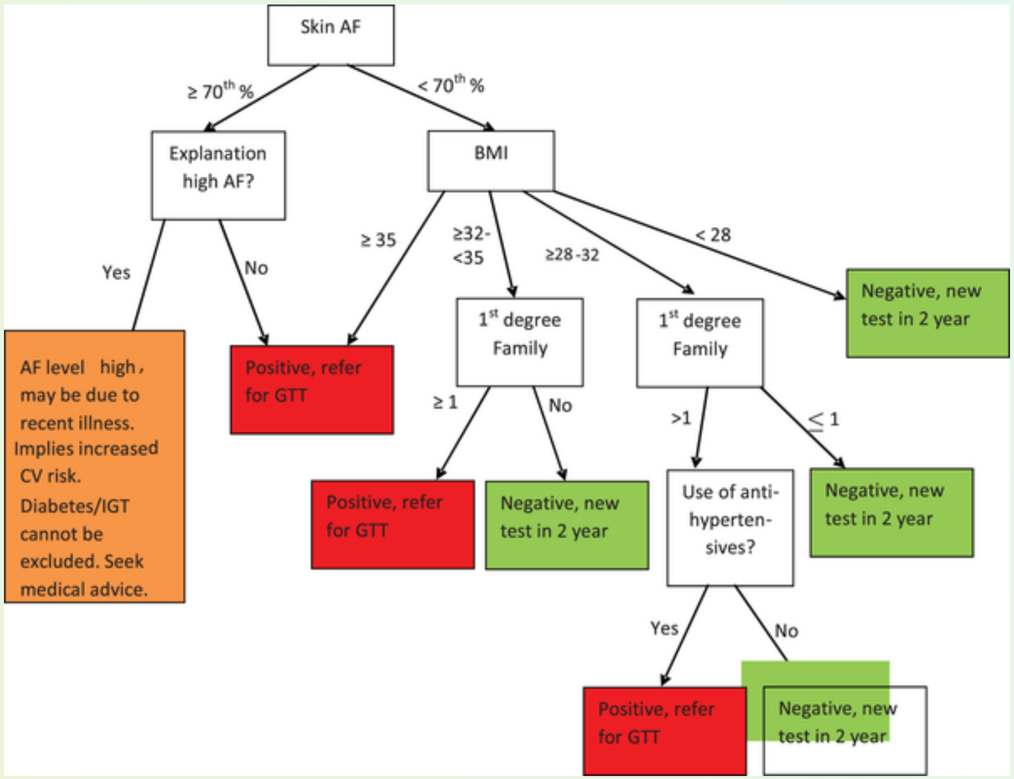
Indukcja drzew

JERZY STEFANOWSKI
Instytut Informatyki
Politechnika Poznańska

Uwagi do wykładu dla Ucz. Maszynowe
Aktualizacja 2016 i 2019

Minimalny plan - czego się nauczyć



Algorithm 1: Buduj_Drzewo $(P, d, \mathbf{T})$

wejście: **P** - Zbiór przykładów, dla których ma być zbudowane drzewo.
d – funkcja decyzyjna
TEST - zbiór możliwych testów.
R_t - dziedzina testu $t \in$ **TEST**.
wyjście: **T** - zbudowane drzewo.
begin
  if *kryterium_stopu*(P, d) **then**
    T.etykieta = kategoria(P, d);
    return;
  t = wybierz_test(**P**, **TEST**);
  T.test = t;
  for $(v \in R_t)$ **do**
    $P_v = \{x \in P | t(x) = v\}$;
    utwórz_nowe_poddrzewo T';
    T.gałąź(v) = T';
    buduj_drzewo (P_v, d, T');
end

Drzewa klasyfikacyjne – algorytm ID3

Podstawowe rozszerzenia – C4.5

Przeuczenie klasyfikatora

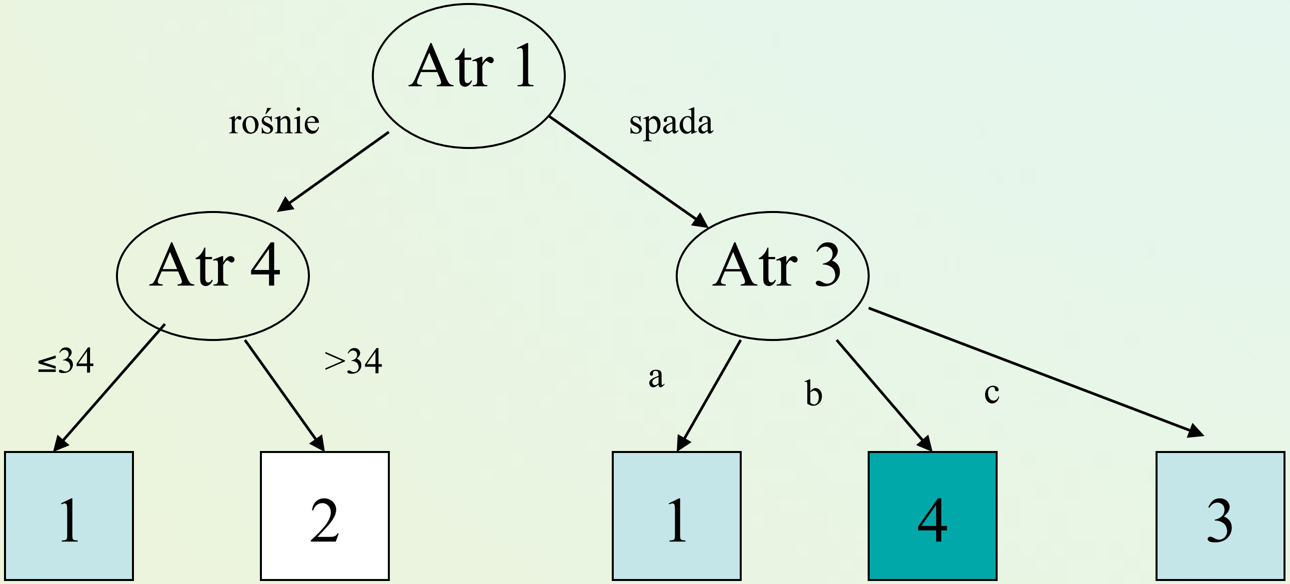
Plan wykładu

1. Drzewa decyzyjne
2. Algorytm ID3, entropia informacji
3. Uwzględnianie niedoskonałych danych
4. Przeuczenie (overfitting)
   - Redukcja drzew - prunning
5. Inne rozszerzenia → C4.5
6. Wybrane zastosowania oraz narzędzia
7. 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 klasyfikacyjne i … (regresji …)

80's

ID3 **CHAID** **CART**

…

90

C4.5 Oblique trees

Ensembles

Bagging **Boosting**

Mega-induction,
Sampling, splits

Random Forests

Ivotes

end. 90

KDD generalizations

Data pre-sorting / reduce disc acc.

Approx. representations
Samples, Histograms

Rotation Forests

Gradient Boosted Machines

SLIQ
SPRINT
…

BOAT
CLOUDS
SPIES

Parallel versions

Incremental trees
ID4, ID5R, ITL

Massive incremental tress
Hoeffding trees, **VFDT**

Stream ensembles
Online …/ block based

MOA,
SAMOA

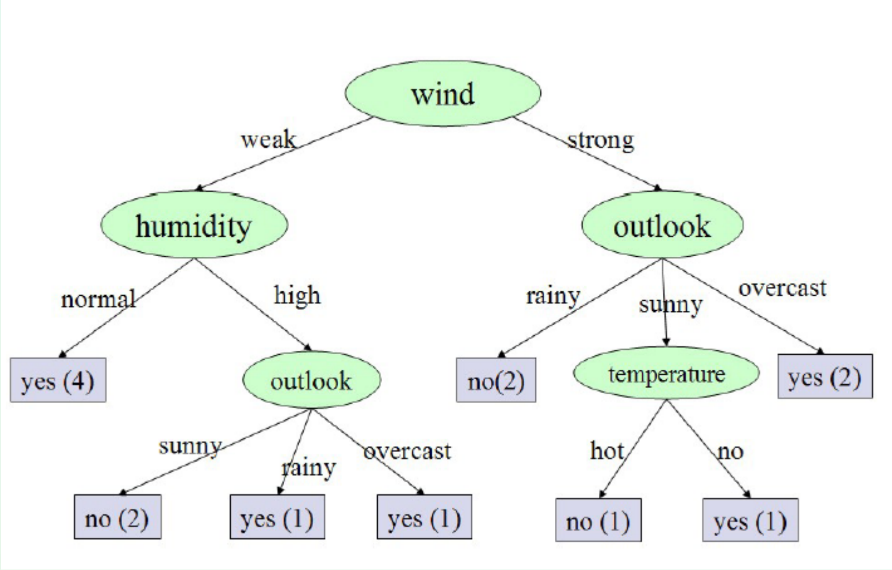
Data streams + drifts
VFDT -> CVFDT; **VFDTc**

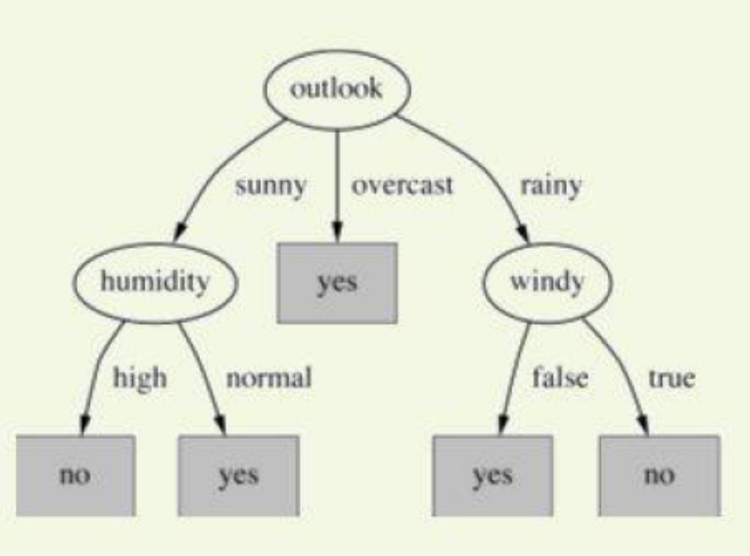
Rozwój implementacji w MapReduce, Hadoop / SPARK

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





Jak poszukiwać drzew?

- Rozważ wszystkie możliwe testy
- Duża liczba hipotez, dla N binarnych atrybutów:
  - 1 null tree
  - N trees with 1 (root) test
  - N*(N-1) trees with 2 tests
  - N*(N-1)*(N-1) trees with 3 tests
  - i dalej, …
- Przestrzeń przeszukiwania rośnie wykładniczo od rozmiaru liczby atrybutów
  - Podejścia heurystyczne

Learning Decision Trees

Decision trees provide a very popular and efficient hypothesis space.

- **Variable Size**. Any boolean function can be represented.
- **Deterministic**.
- **Discrete and Continuous Parameters**.

Learning algorithms for decision trees can be described as

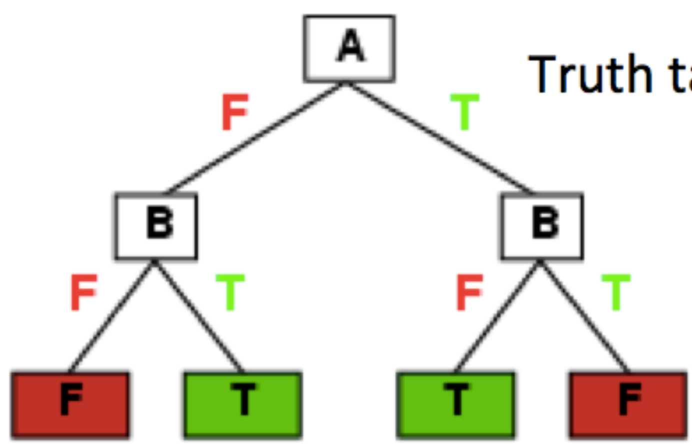
- **Constructive Search**. The tree is built by adding nodes.
- **Eager.**
- **Batch** (although online algorithms do exist).

Decision Trees Provide Variable-Size Hypothesis Space

As the number of nodes (or depth) of tree increases, the hypothesis space grows

- **depth 1** ("decision stump") can represent any boolean function of one feature.
- **depth 2** Any boolean function of two features; some boolean functions involving three features (e.g., $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_3)$)
- **etc.**

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

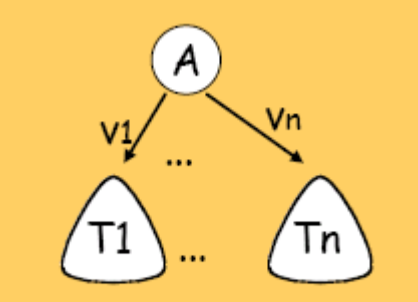


Truth table row → path to leaf

Uwagi ogólne

- Heurystyczny algorytm rekurencyjnego podziału zbioru przykładów uczących
  - TDIDT → Top Down Induction of Decision Trees.
- Główne zagadnienia:
  - **Splitting criterion**: kryterium podziały / jak wybrać najlepszy test w węźle
  - **Stopping criterion**: kiedy wstrzymać rozbudowę gałęzi drzewa
  - **Pruning**: upraszczanie struktury drzewa, aby polepszać zdolności uogólnienia / klasyfikacji nowych faktów
- Inne:
  - Uwzględnianie atrybutów liczbowych, nieznanych wartości, ….

Metody indukcji drzew decyzyjnych



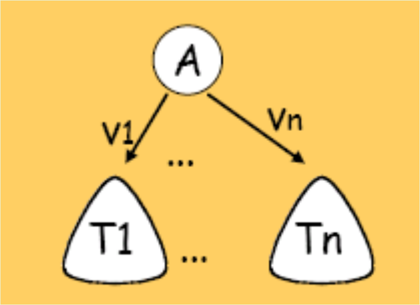
- Podejście obejmuje dwa etapy:
  - Konstrukcja drzewa (rekurencyjna procedura)
    - Na początku wszystkie przykłady w węźle.
    - Rekurencyjnie dziel przykłady w oparciu o wybrane testy na wartościach atrybutu (kryterium wyboru najlepszego atrybutu).
    - Zatrzymaj gdy wszystkie przykłady „w gałęzi" 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,…

Ogólny schemat ID3

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)});

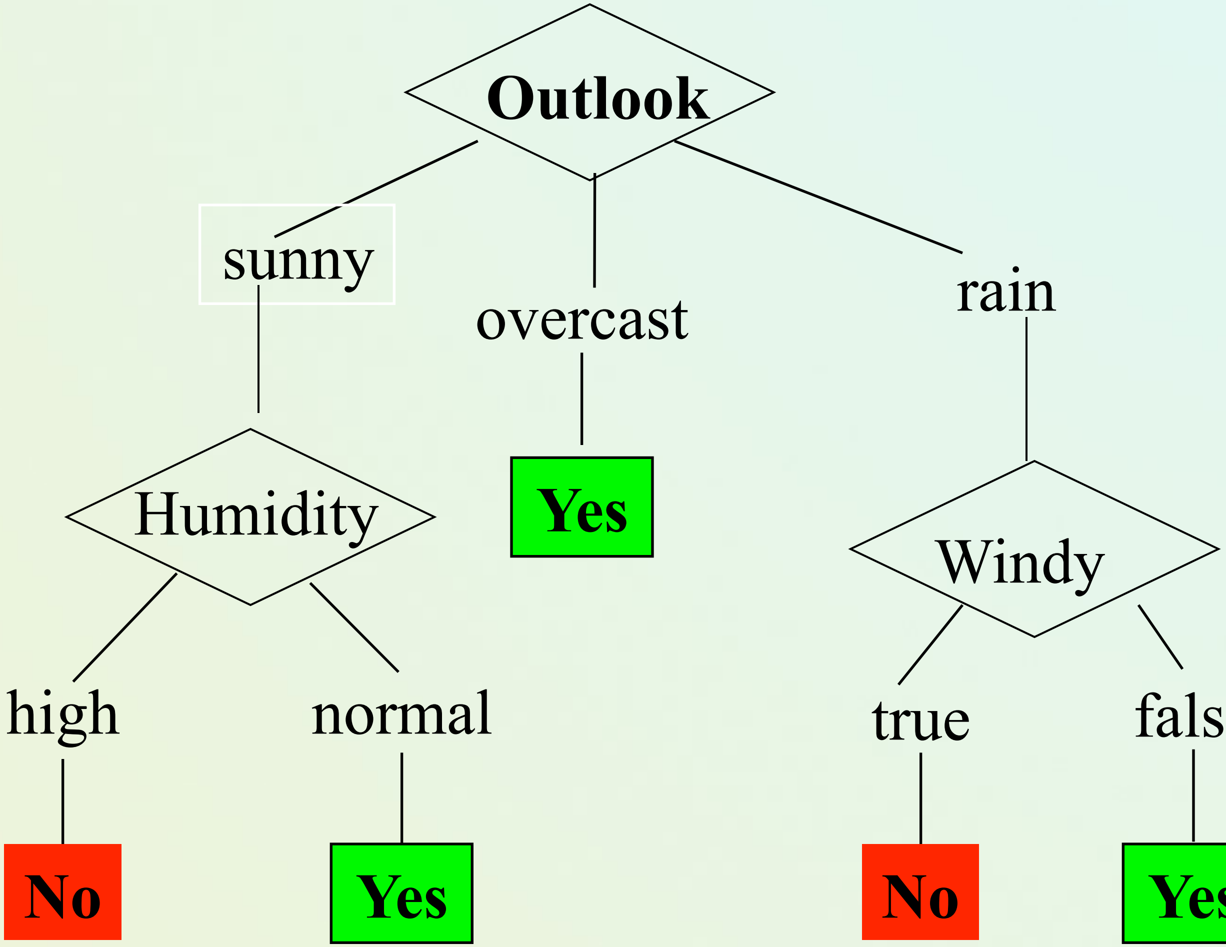


Weather Data: Play or not Play?

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

Wszystkie atrybuty – są nominalne

Example Tree for “Play?”

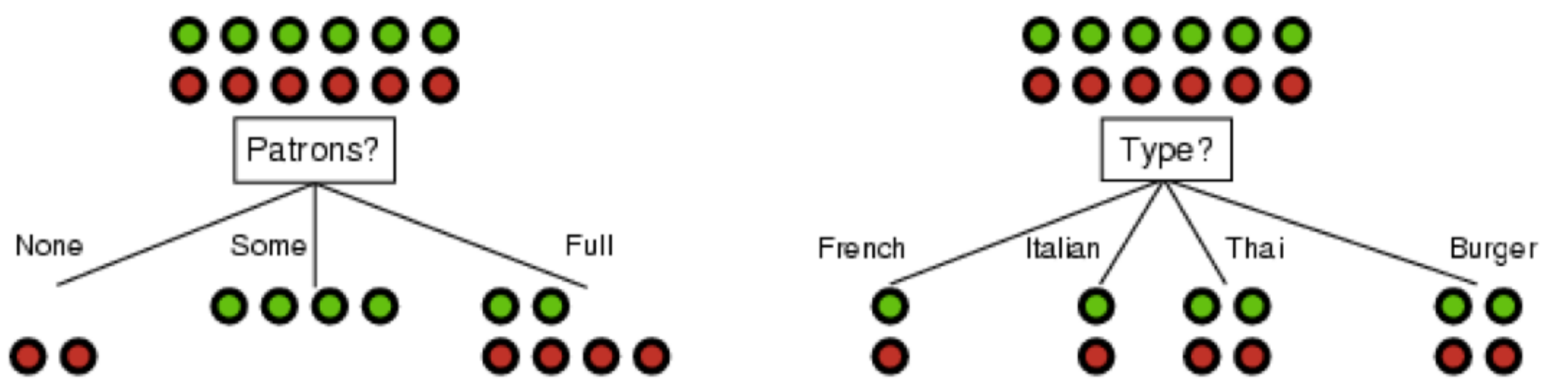


Intuicja wyboru atrybutu

Przykład decyzji o wyborze restauracji [Rusell, Norvig]

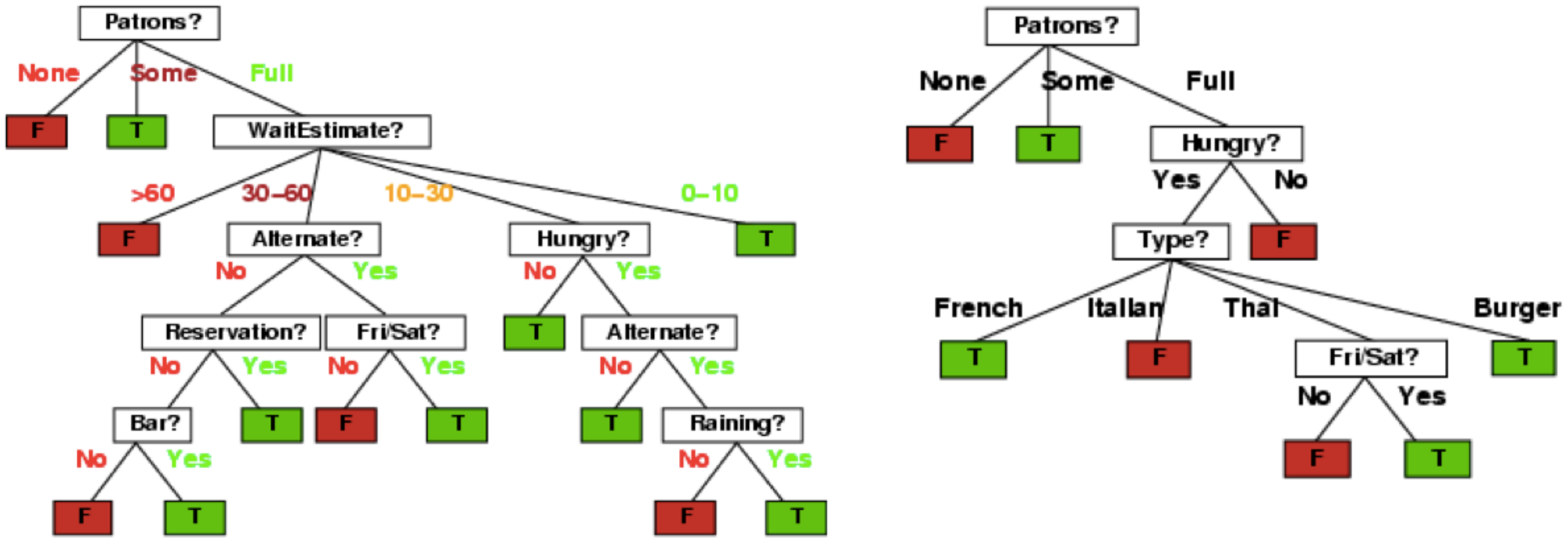
Split condition -

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



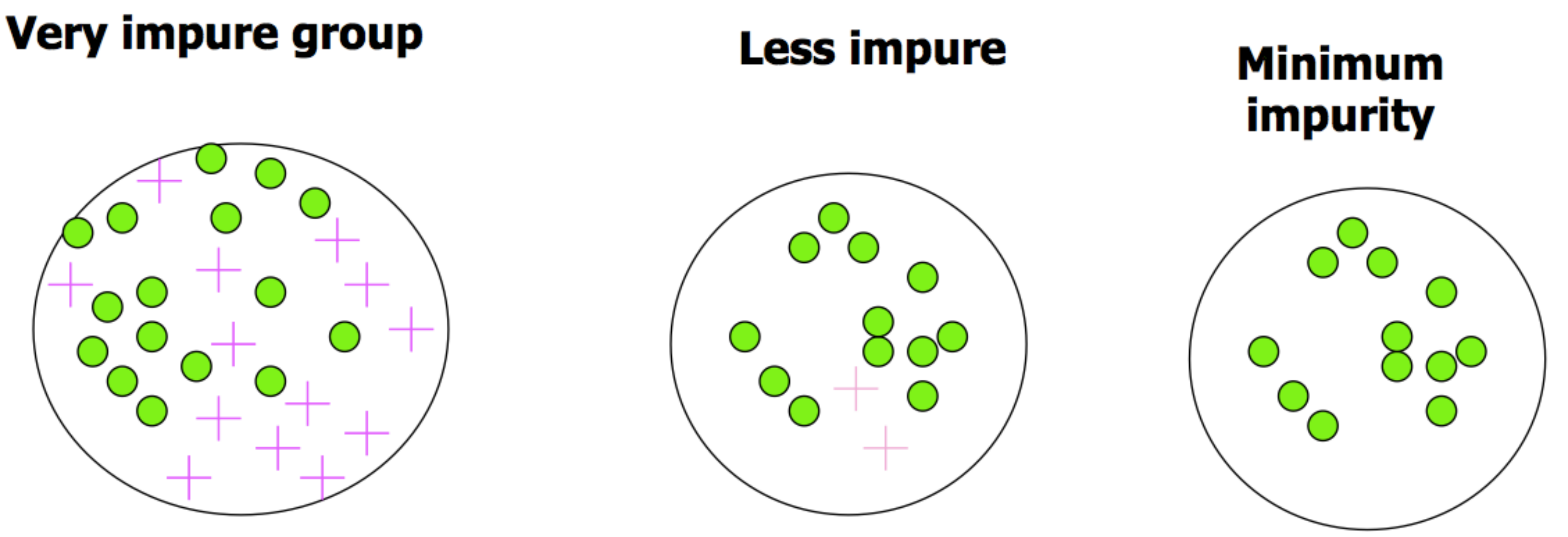
Which split is more informative: *Patrons?* or *Type?*

Dążenie do najprostszego drzewa



Based on Slide from M. desJardins & T. Finin

Impurity functions



Intuicja funkcji “impurity class assignments”

Za wykładem Eric Eaton

Oczekiwania wobec funkcji wyboru atrybutu

Impurity functions:

- Given a random variable *x* with *k* discrete values, distributed according to P={p1,p2,…pk}, a impurity function Φ should satisfies:
  - $\Phi(P) \geq 0$; $\Phi(P)$ is minimal if $\exists i$ such that pi=1;
    $\Phi(P)$ is maximal if $\forall i$ $1 \leq i \leq k$, pi=1/k
    $\Phi(P)$ is symmetrical and differentiable everywhere in its range
- The goodness of split is a reduction in impurity of the target concept after partitioning S.
- Popular function: *information gain*
  - Information gain increases with the average purity of the subsets that an attribute produces

Wybór atrybutu

Który atrybut może tworzyć dobry podział

p_+ i p_- - proporcje w lewej i prawej gałęzi.

Zbiór przykładów *S*

Ile informacji zawiera dany podział ?
Średnia l. bitów do zakodowania dowolnego przykładu z *S* wynosi:

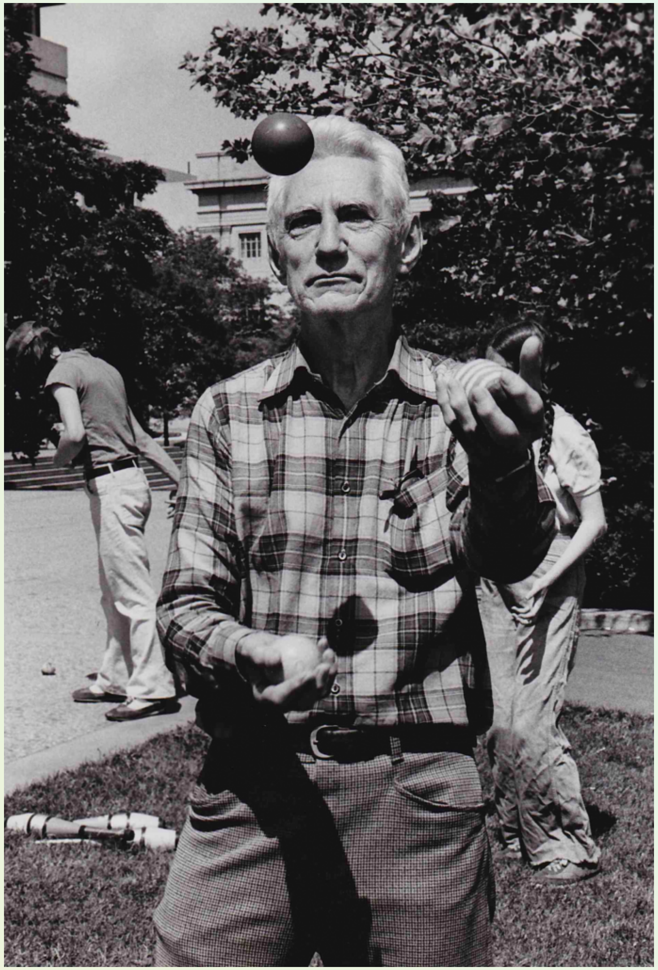
$$\mathrm{entropy}(S) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

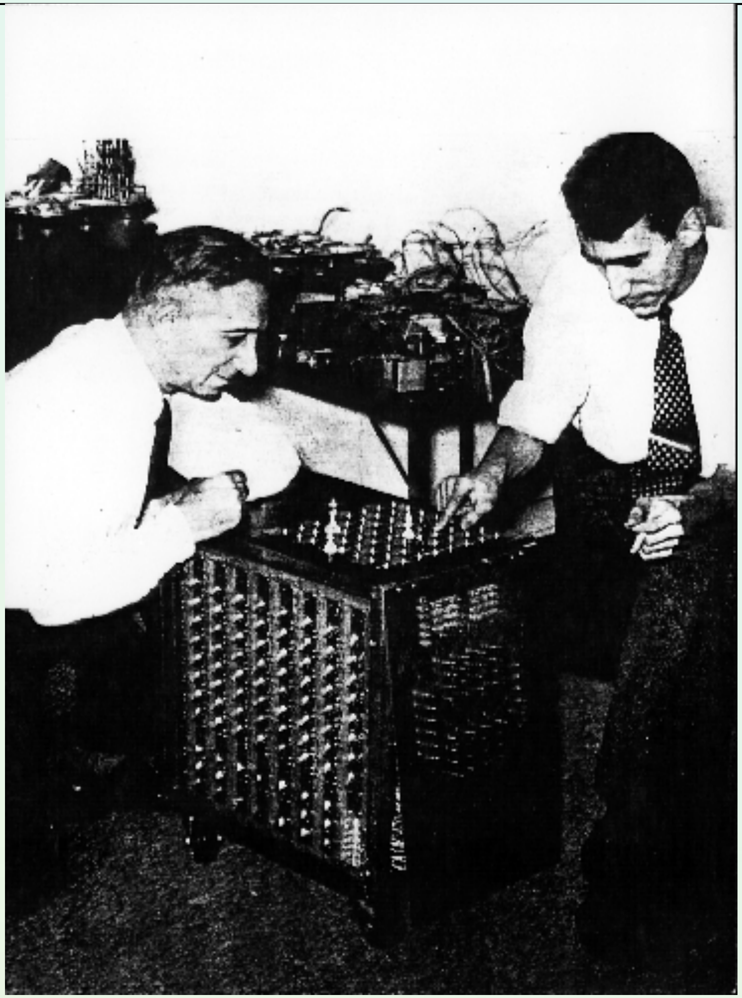
$$entropy(S \mid A) = \sum_{i=1}^{m} \frac{|S_i|}{|S|} \cdot entropy(S)$$

Informacja dla czystych węzłów = 0;

jest max dla najbardziej pomieszanych.

Przy okazji – poczytaj o Claude Shannon

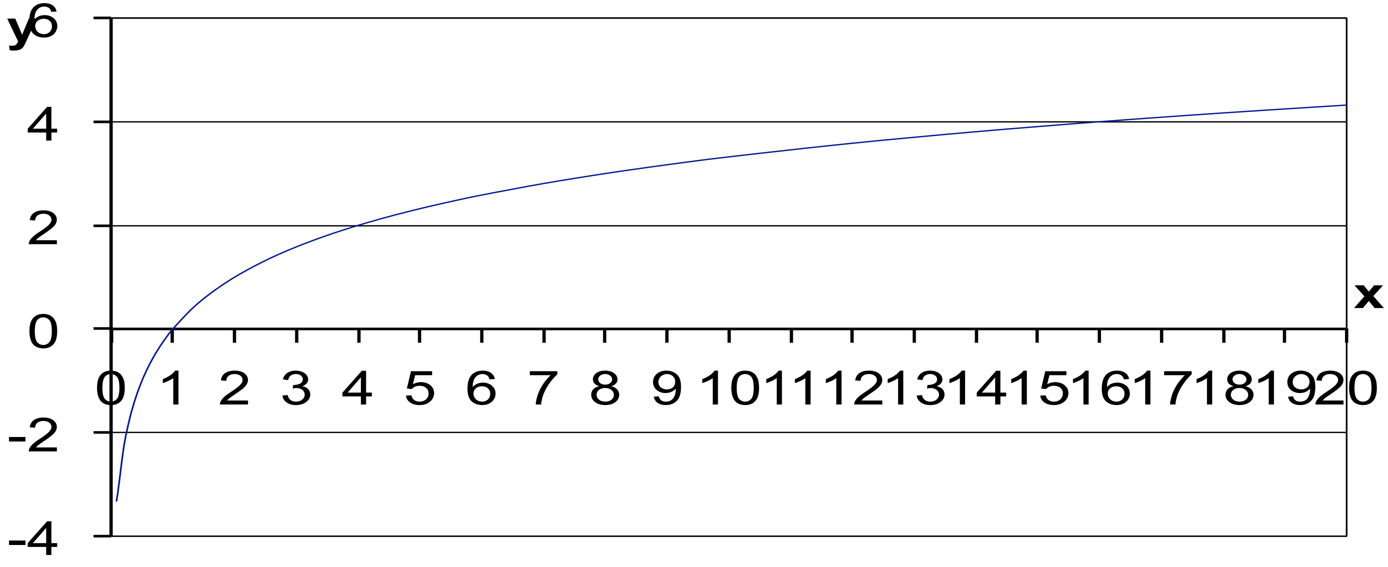




- If we were fish, we would think of information as our water. We are in it all the time. We take it in and we breathe it out.

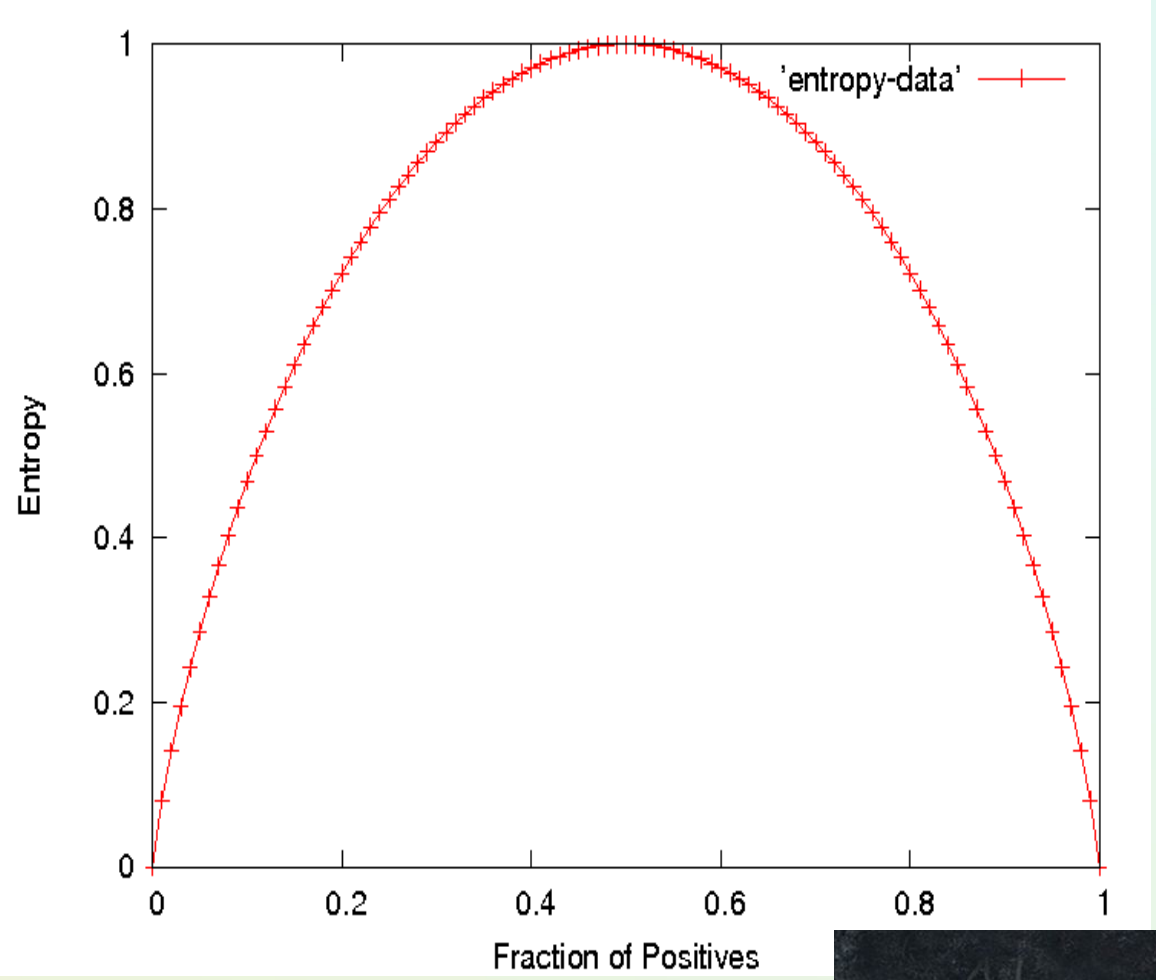
Przypomnienie logarytmów

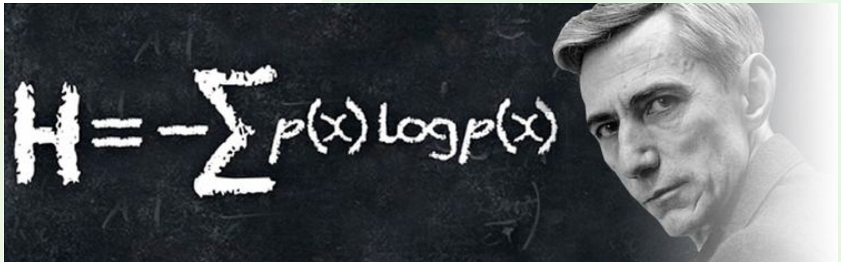
- Funkcja log. $y = \log_a x$
- a – podstawa logarytmu $x = a^y$
- Rozważmy funkcję logarytmiczną dla a = 2 (tj. $\log_2 x$)



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

Entropy Plot for Binary Classification





Weather Data: Play or not Play?

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

Note:
All attributes are nominal

Entropia dla przykładu Quinlana 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*

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

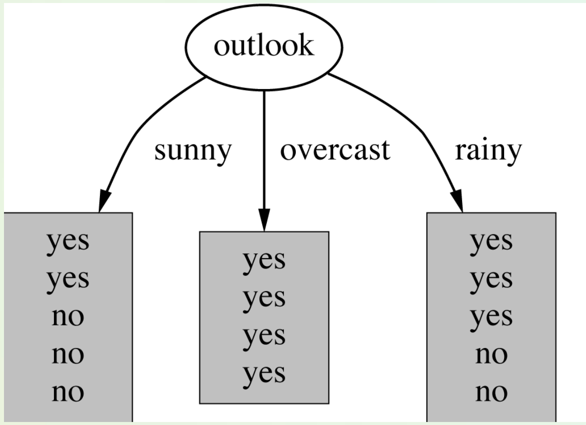
$$p_{no} = -\left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) = 0.53$$

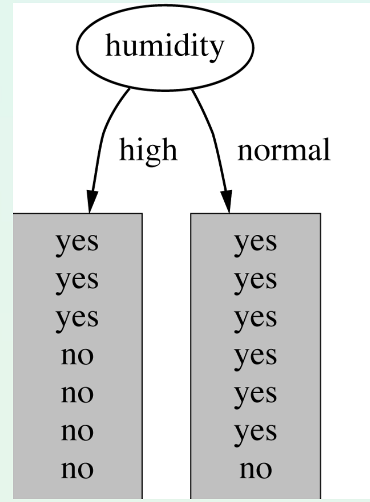
$$E(S) = p_{yes} + p_{no} = 0.94$$

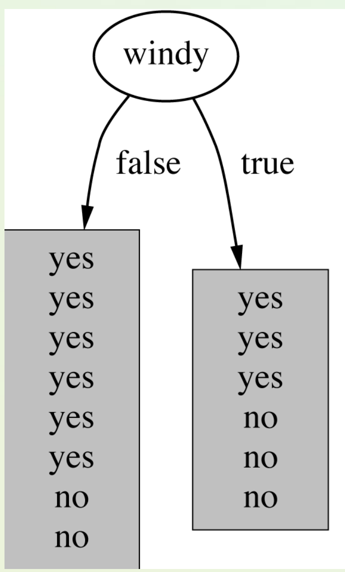
Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes

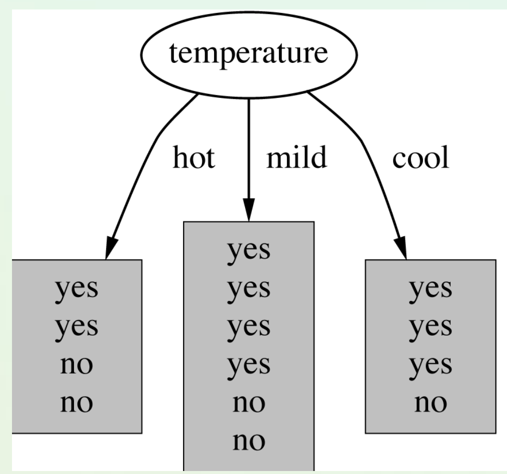
Outlook	Temp.	Humidity	Windy	play
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Korzeń drzewa – który atrybut









witten&eibe

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:

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

Wybierz atrybut “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”:

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

Note: log(0) is not defined, but we evaluate 0*log(0) as zero

- “Outlook” = “Rainy”:

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

- Expected information for attribute:

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

$$= 0.693$$

Oblicz zysk informatyczny dla każdego atrybutu

- Information gain:

(information before split) – (information after split)

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

$$= 0.247$$

- Information gain for attributes from weather data:

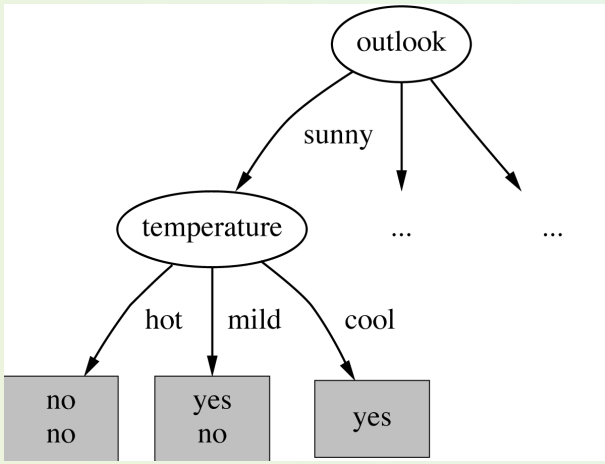
$$\text{gain}(\text{"Outlook"}) = 0.247$$

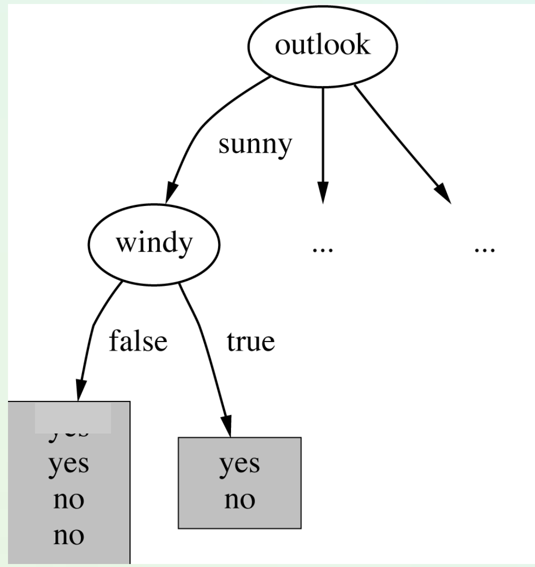
$$\text{gain}(\text{"Temperature"}) = 0.029$$

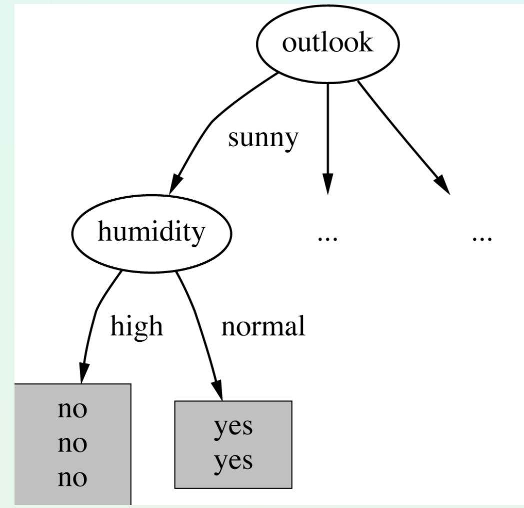
$$\text{gain}(\text{"Humidity"}) = 0.152$$

$$\text{gain}(\text{"Windy"}) = 0.048$$

Rozbuduj drzewo





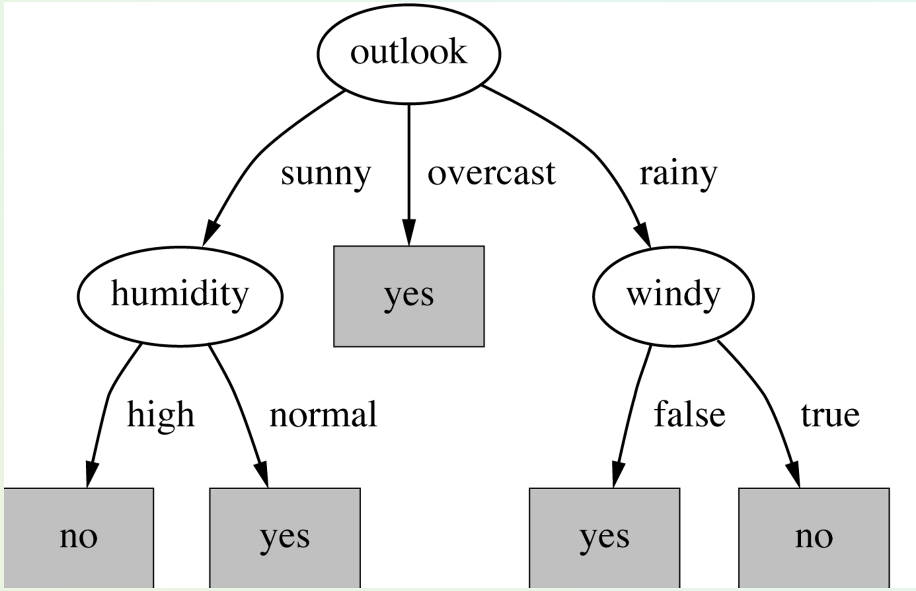


$$\text{gain}(\text{"Humidity"}) = 0.971$$

$$\text{gain}(\text{"Temperature"}) = 0.571$$

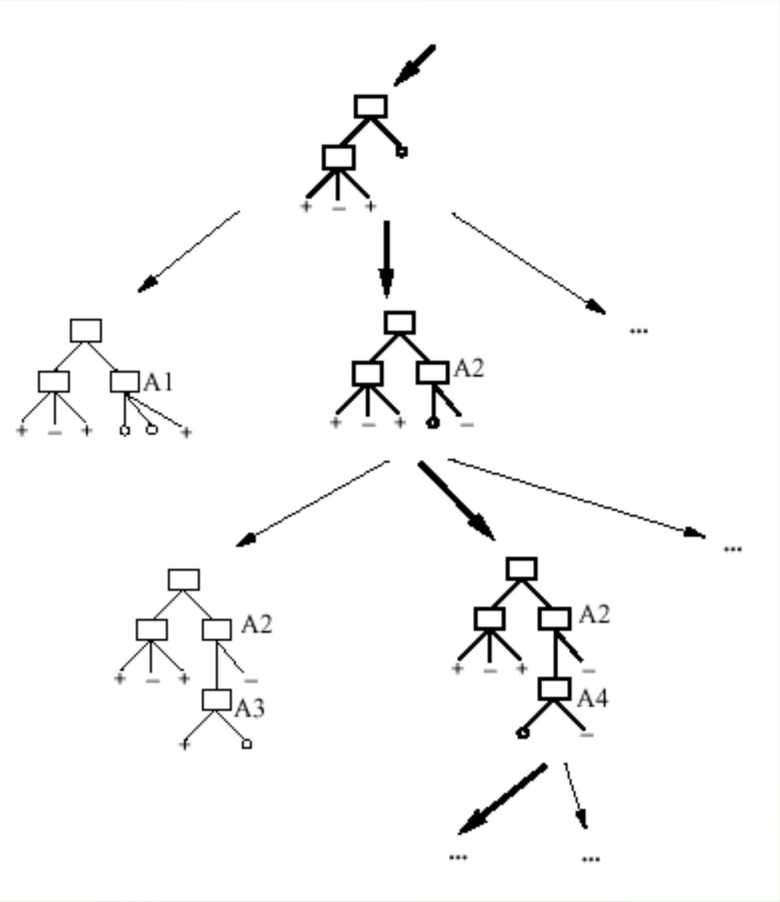
$$\text{gain}(\text{"Windy"}) = 0.020$$

Końcowe drzewo



ID3 - Quinlan

Tworzenie drzewa – przeszukiwanie przestrzeni



Tworzenie drzewa: szukanie w przestrzeni hipotez;

Kryterium wyboru atrybutu dość odporne na ”zaburzenia" danych wejściowych

ID3 - podział w oparciu o **zysk informacyjny**.

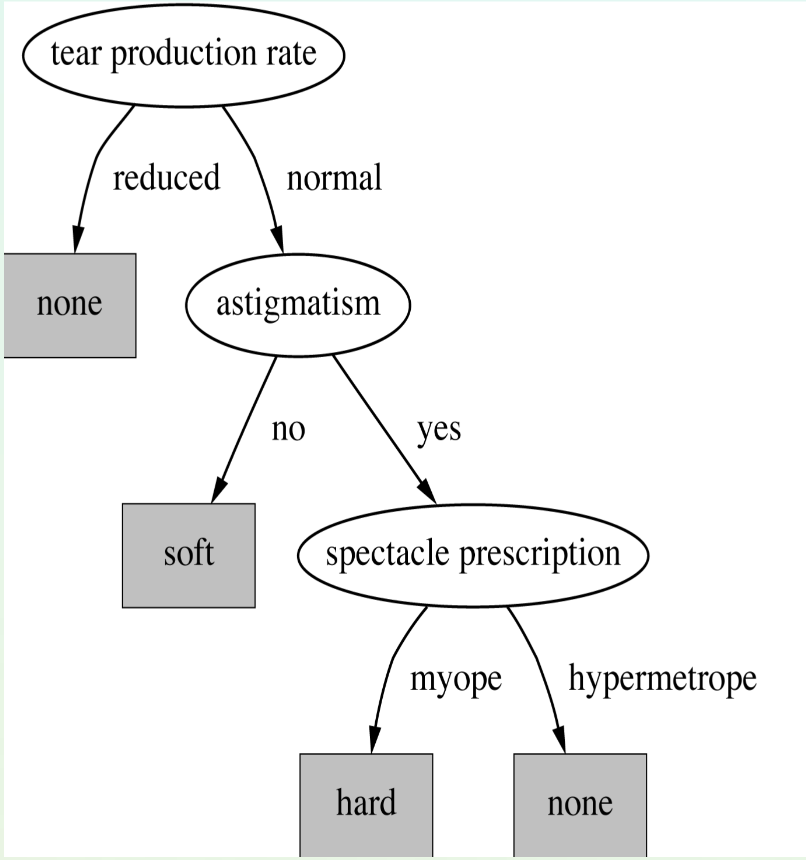
Lepsze mniejsze drzewo.

Dość odporne na „szum.”

Lokalne minima – strategia bez nawrotów.

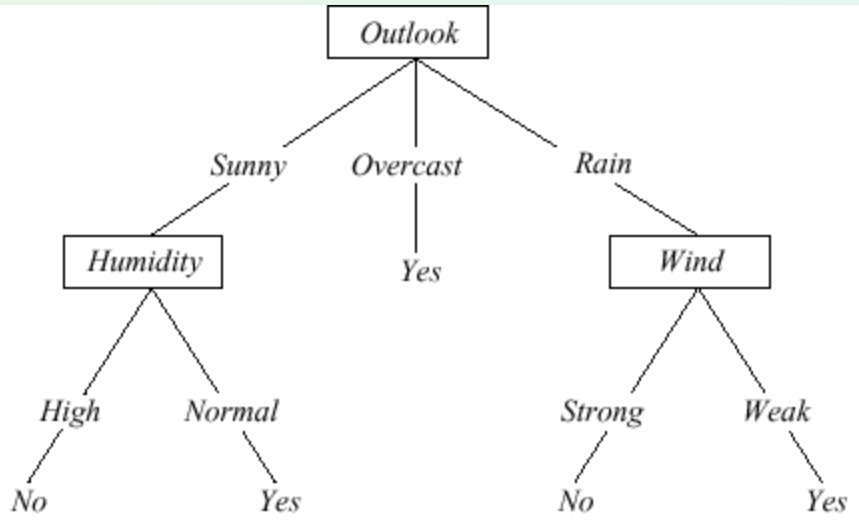
Wykorzystanie drzewa

- 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 $PlayTennis = Yes$

Zamiana na zbiór reguł klasyfikacyjnych

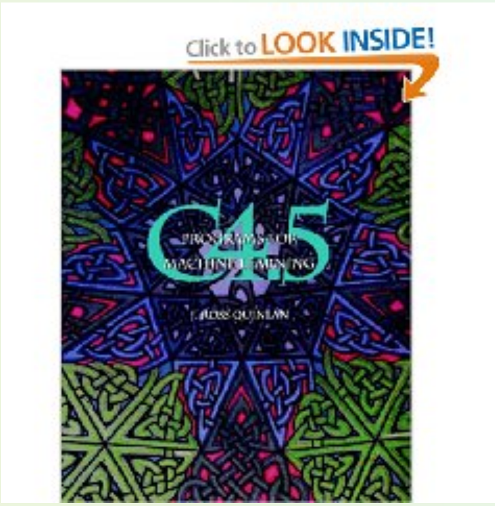
Poprzedni przykład:

```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```

Lecz pamiętaj:

- Dropping redundant conditions in rules and rule post-pruning
- Classification strategies with rule sets are necessary

J.Ross Quinlan

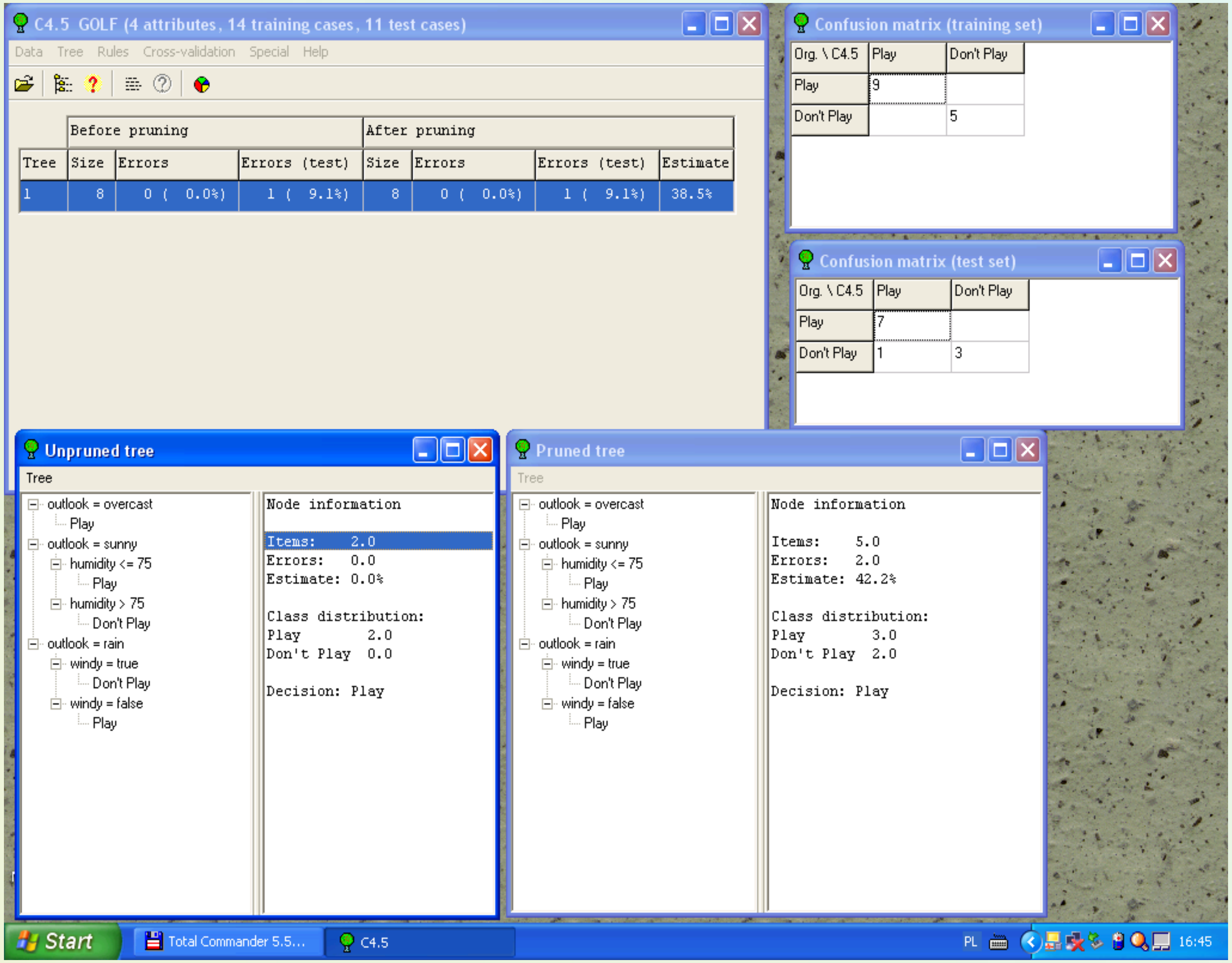


Ross Quinlan completed a PhD in Computer Science at the University of Washington in 1968. He has developed several algorithms used in machine learning and data mining such as ID3, C4.5, FOIL, and more recent commercial systems such as See5 and Cubist. He has held permanent appointments at the University of Sydney, University of Technology Sydney, Rand Corporation, and visiting appointments at Carnegie-Mellon University, MIT, GTE, and Stanford University. He currently heads a small data mining tools company and is an Adjunct Professor at the University of New South Wales. He is a Fellow of the American Association for Artificial Intelligence and the Australian Computer Society.

- More of his papers – have a look at http://www.rulequest.com/Personal/
- See also http://en.wikipedia.org/wiki/Ross_Quinlan

C4.5 Quinlan + PP interfejs

- Opiera się na oryginalnym kodzie J.R. Quinlana



Weka J48 Trace

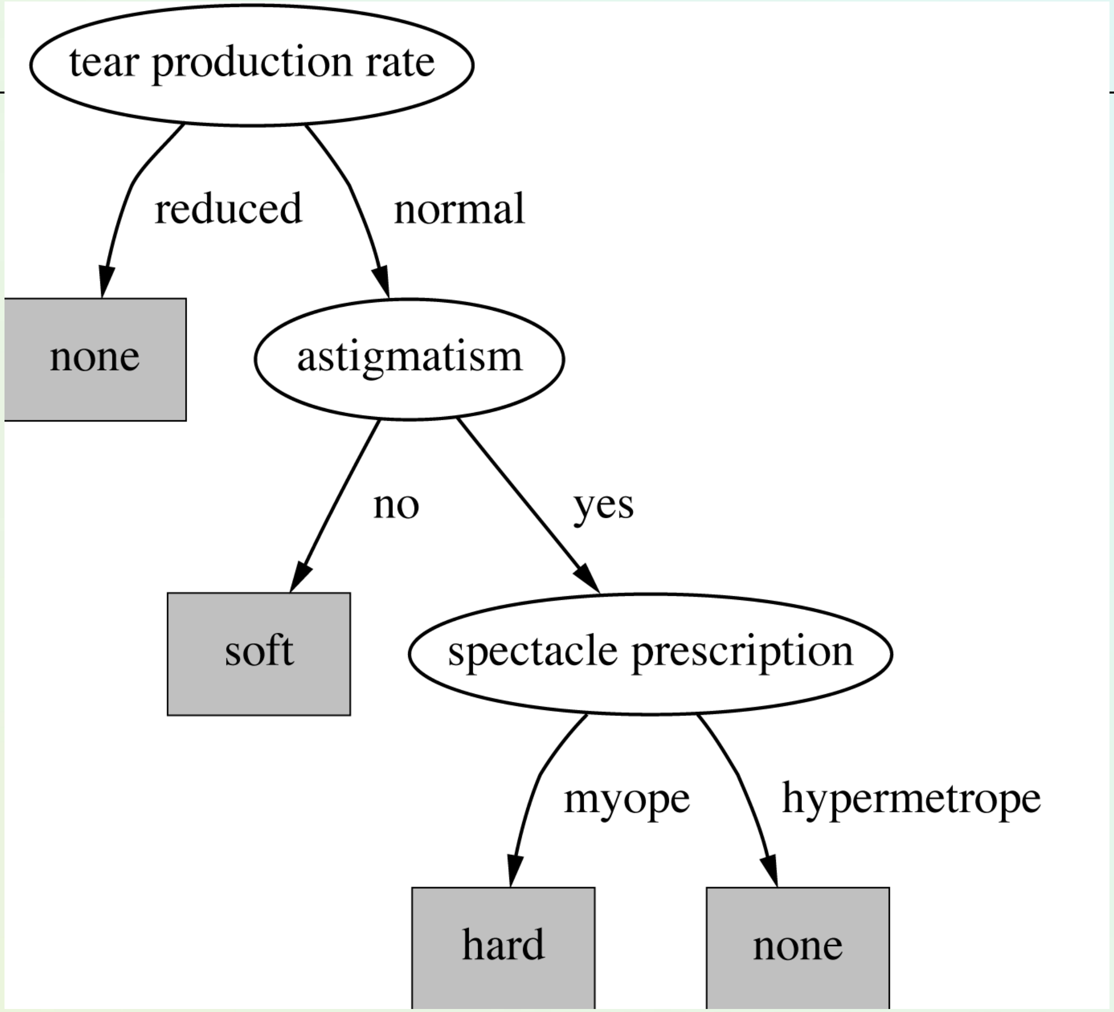
```
data> java weka.classifiers.trees.J48 -t contact-lenses.arff
J48 pruned tree
------------------
tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no: soft (6.0/1.0)
|   astigmatism = yes
|   |   spectacle-prescrip = myope: hard (3.0)
|   |   spectacle-prescrip = hypermetrope: none (3.0/1.0)

Number of Leaves  :     4
Size of the tree :      7

Time taken to build model: 0.03 seconds
Time taken to test model on training data: 0 seconds

=== Error on training data ===

Correctly Classified Instances          22               91.6667 %
Incorrectly Classified Instances         2                8.3333 %
Kappa statistic                          0.8447
Mean absolute error                      0.0833
Root mean squared error                  0.2041
Relative absolute error                 22.6257 %
Root relative squared error             48.1223 %
Total Number of Instances               24
```

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  0  3  1 |  b = hard
  1  0 14 |  c = none


=== Stratified cross-validation ===

Correctly Classified Instances          20               83.3333 %
Incorrectly Classified Instances         4               16.6667 %
Kappa statistic                          0.71
Mean absolute error                      0.15
Root mean squared error                  0.3249
Relative absolute error                 39.7059 %
Root relative squared error             74.3898 %
Total Number of Instances               24


=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  0  3  1 |  b = hard
  1  2 12 |  c = none
```

Reprezentacja przykładów – tablica danych

- Przykłady tzw. contact lenses / dobór szkieł kontaktowych:
- Atrybuty:
    - age {young, pre-presbyopic, presbyopic}
    - spectacle-prescrip {myope, hypermetrope}
    - astigmatism {no, yes}
    - tear-prod-rate {reduced, normal}
- Decyzja contact-lenses {soft, hard, none}

age	specpres	astig	tearprod	contlen
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none



Decision tree = Drzewo decyzyjne / drzewo klasyfikacyjne
Algorytm (ID3/C4.5)

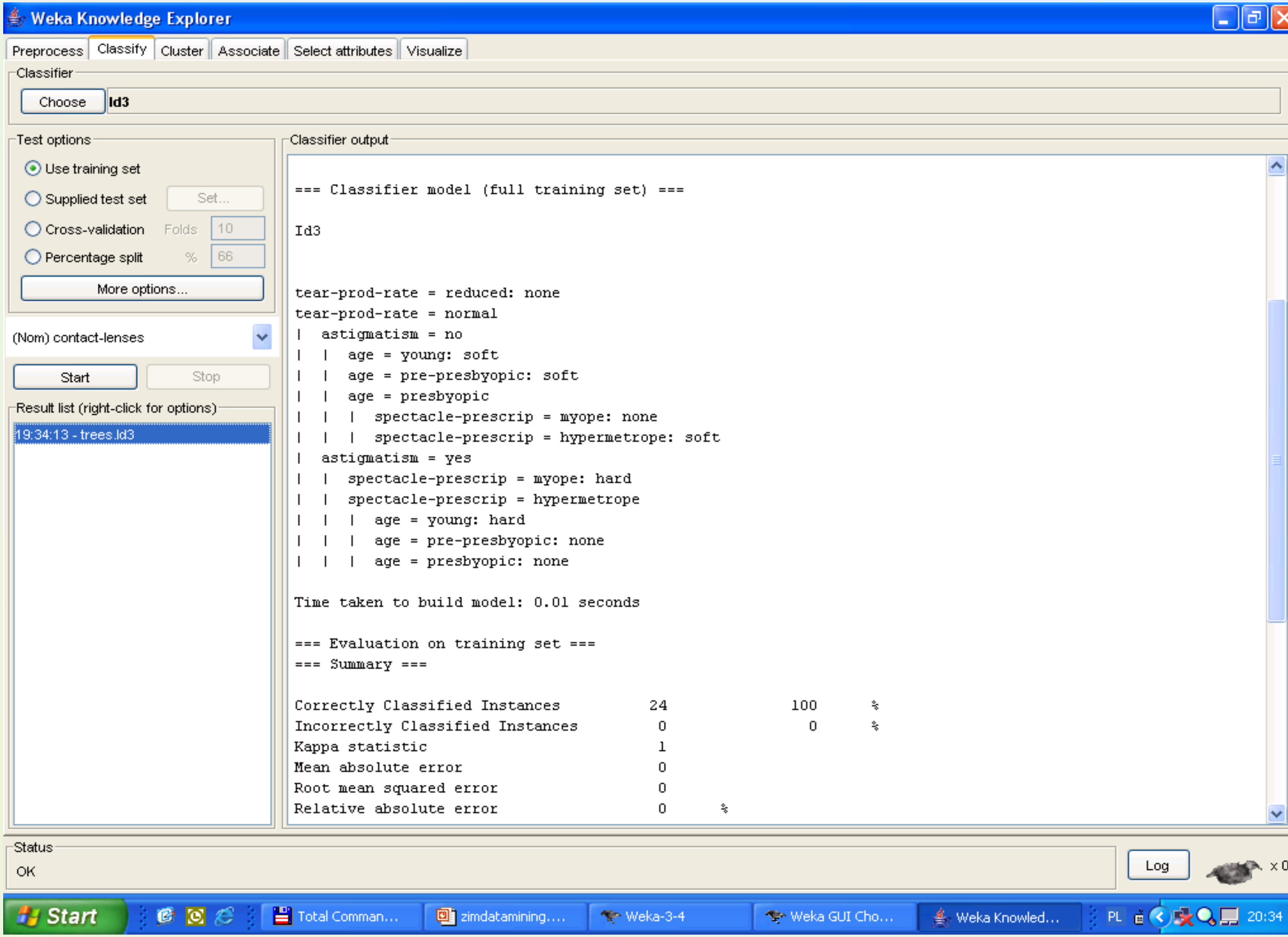
Weka – software for data mining

- Waikato Environment for Knowledge Analysis (WEKA); developed by the Department of Computer Science, University of Waikato, New Zealand
- Data mining / Machine learning software written in Java (distributed under the GNU Public License)
- Used for research, education, and applications

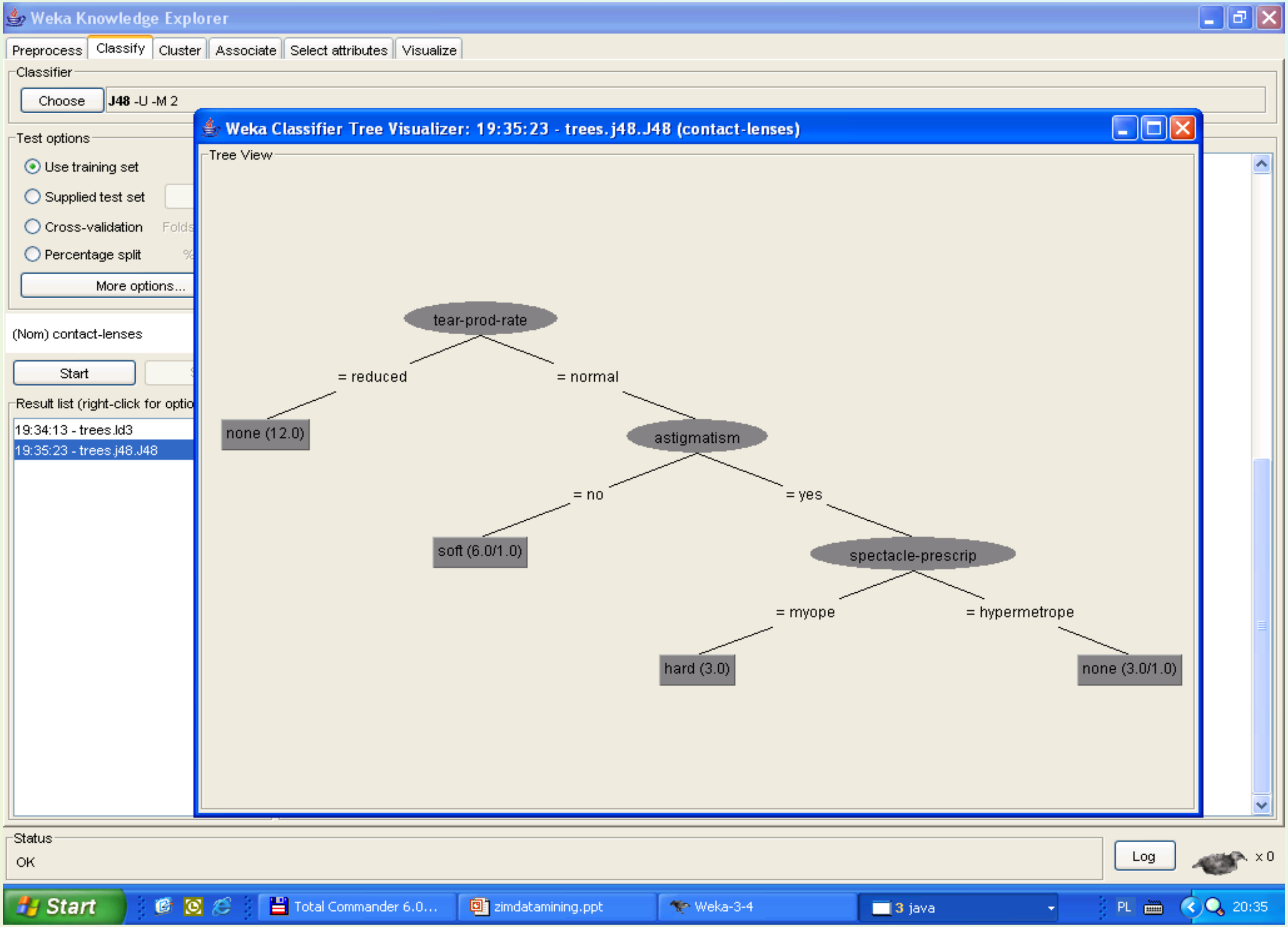
http://www.cs.waikato.ac.nz/ml/weka/

- Ian Witten, Eibe Frank

Poszukiwanie drzew ze zbioru przykładów



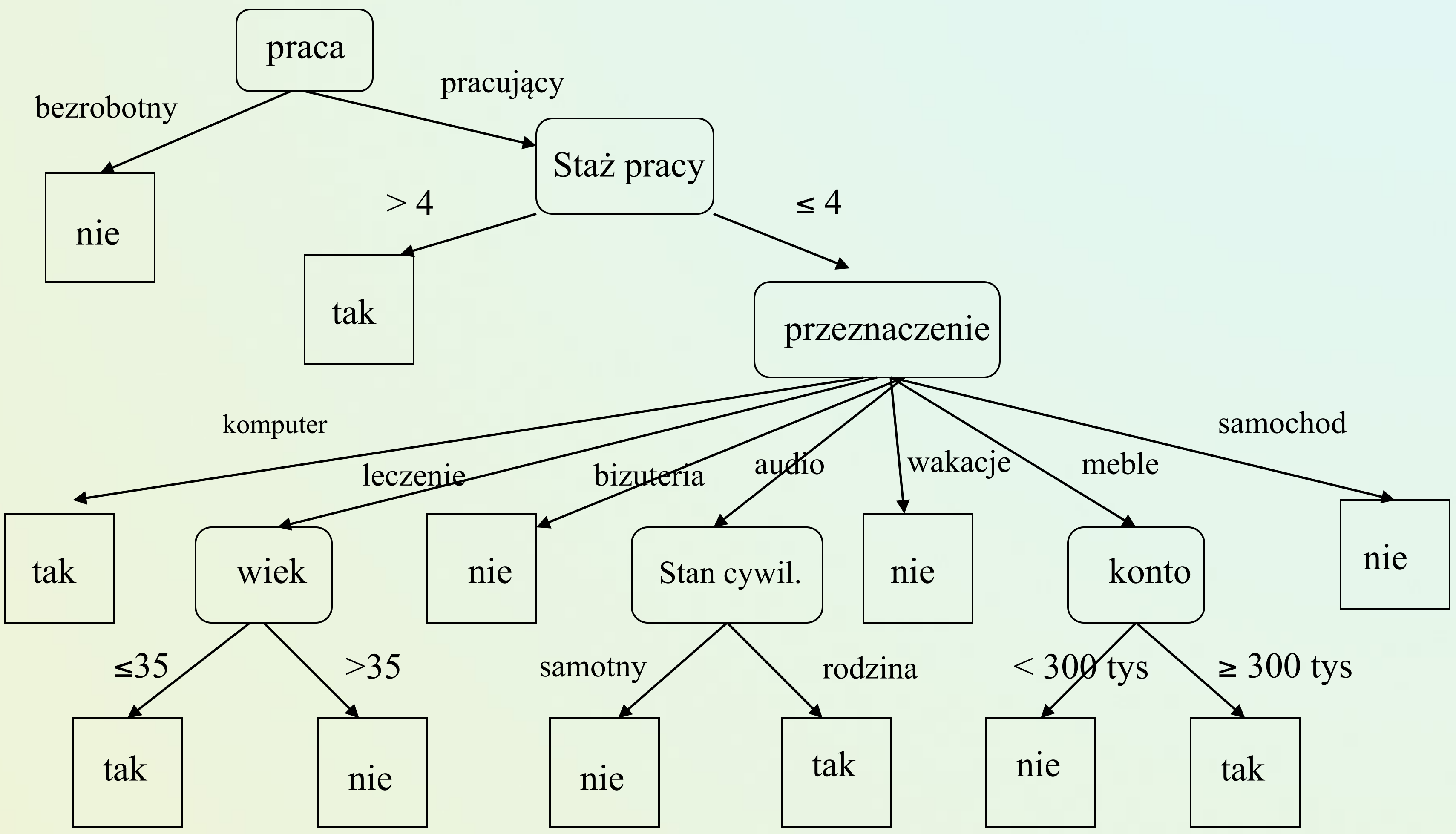
Widok drzewa



Japanese credit data

- Data about 125 customers applying for a loan in one bank (repository of Univ. Irvine).
- Person characterized by 10 attributes, e.g. profession, aim of the loan, martial status, age, income, account status, month payment of interest, ...
- Customers: good (return of the credit with interests) and risk (stop paying).
- Aim of the analysis: find rules / profiles identifying risky customers.

Final tree for credit data



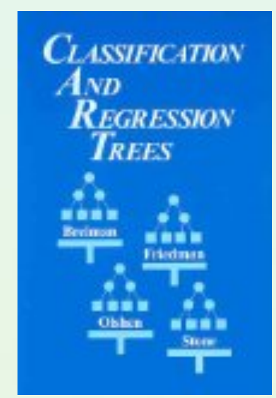
Other splitting criteria - rozszerzania

- Gini index (CART, SPRINT)
  - select attribute that minimize impurity of a split
- $\chi 2$ contingency table statistics (CHAID)
  - measures correlation between each attribute and the class label
  - select attribute with maximal correlation
- Normalized Gain ratio (Quinlan 86, C4.5)
  - normalize different domains of attributes
- Distance normalized measures (Lopez de Mantaras)
  - define a distance metric between partitions of the data.
  - chose the one closest to the perfect partition
- Orthogonal (ORT) criterion
- AUC-splitting criteria (Ferri et at.)
- There are many other measures. Mingers'91 provides an experimental analysis of effectiveness of several selection measures over a variety of problems.
- Look also in a study by D.Malerba, …

CART – Classification And Regression Tree

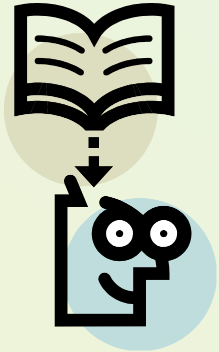
- Developed 1974-1984 by 4 statistics professors
  - Leo Breiman (Berkeley), Jerome Friedman (Stanford), Charles Stone (Berkeley), Richard Olshen (Stanford)
- Focused on accurate assessment when data is noisy
- Currently distributed by Salford Systems



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
- Oraz czytamy książki anglojęzyczne



Pytanie i komentarze?

?