# Data Mining

Pertemuan 5: Classification
(lanjutan)

# Review Kuliah Sebelumnya

- Prosedur umum untuk membangun sebuah model classifier:
  - Mempartisi data menjadi Training dan Testing
  - Melatih classifier
  - Mengetes classifier sebelum digunakan:
    - Ukur kinerjanya dengan Precision, Recall, F-Measure
  - Mengkombinasikan beberapa classifier
    - Majority Vote
    - Linear Weight Combination
  - Binary-class vs Multi-class

# Algoritma Klasifikasi

- Induksi pohon keputusan (*decision tree*)
- Bayesian Classification
- Neural Network Classification
- Support Vector Machine Classification

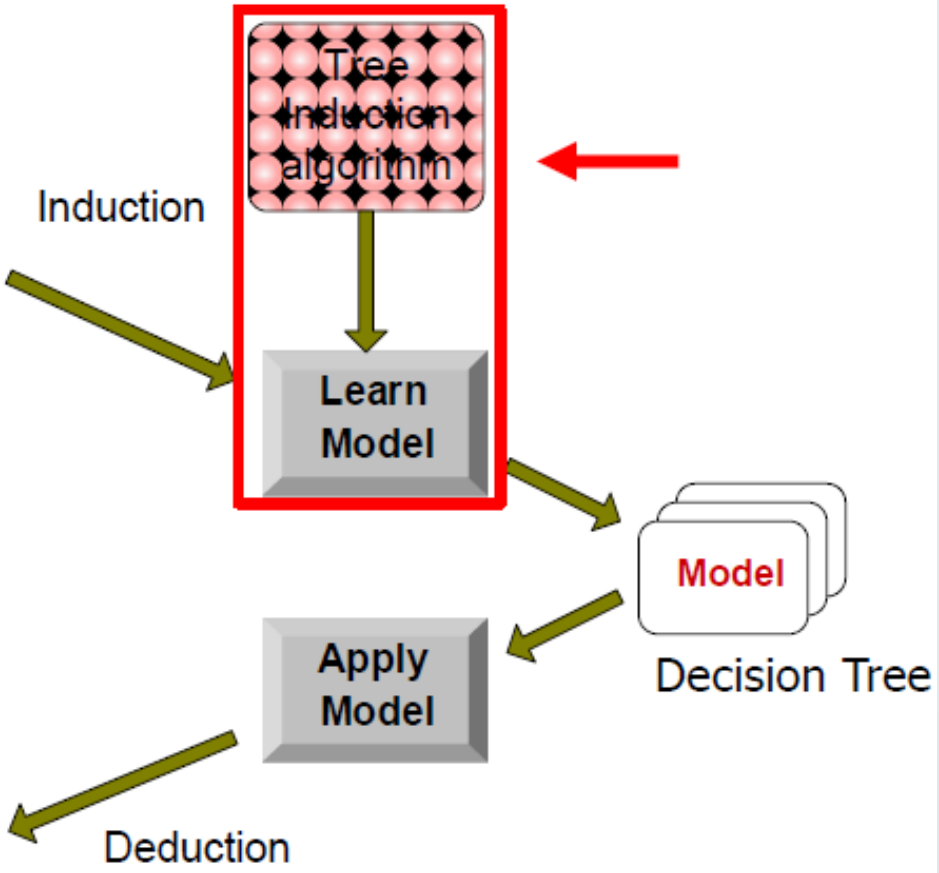
# Klasifikasi Pohon Keputusan

| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 1 | Yes | Large | 125K | No |
| 2 | No | Medium | 100K | No |
| 3 | No | Small | 70K | No |
| 4 | Yes | Medium | 120K | No |
| 5 | No | Large | 95K | Yes |
| 6 | No | Medium | 60K | No |
| 7 | Yes | Large | 220K | No |
| 8 | No | Small | 85K | Yes |
| 9 | No | Medium | 75K | No |
| 10 | No | Small | 90K | Yes |

Training Set

| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|---|---|---|---|---|
| 11 | No | Small | 55K | ? |
| 12 | Yes | Medium | 80K | ? |
| 13 | Yes | Large | 110K | ? |
| 14 | No | Small | 95K | ? |
| 15 | No | Large | 67K | ? |

Test Set



# Apa Itu Pohon Keputusan?

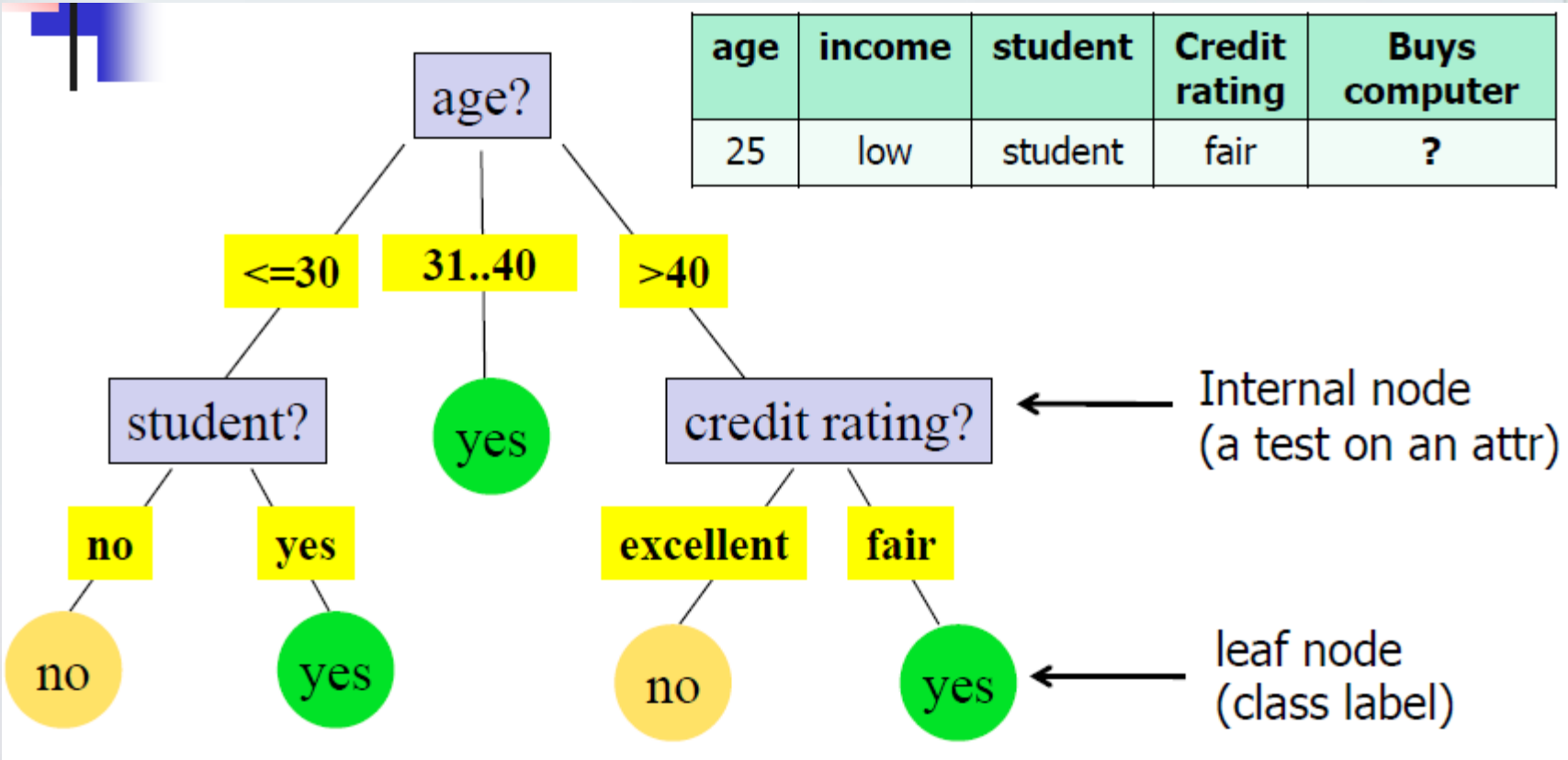
- Struktur tree seperti flowchart
- Node internal: test/cek pada sebuah atribut
  - Contoh: usia > 30?
- Node eksternal/daun: label class
  - Membeli komputer: YA atau TIDAK

# Contoh

- Menentukan apakah seorang pelanggan akan membeli komputer atau tidak

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# Pohon Keputusan untuk Pembelian Komputer




| age | income | student | Credit rating | Buys computer |
|---|---|---|---|---|
| 25 | low | student | fair | ? |

# Induksi Pohon Keputusan

- Algoritma:
  - Hunt's Algorithm
  - CART
  - ID3 (Iterative Dichotomizer 3), C4.5, C5
  - SLIQ, SPRINT
- Struktur umum:
  - Pemilihan atribut dengan Entropi Informasi
  - Induksi pohon keputusan
  - Pembuatan aturan *(rule)*

# Pembuatan Aturan

- Aturan didasarkan hanya pada sampel input
- Aturan seharusnya dapat digunakan untuk memprediksi hasil pada kasus lain (yang bukan sampel input)
- Jika prediksi tidak dapat dilakukan, maka aturan perlu diubah, baik secara otomatis atau manual

# Rumusan Permasalahan

- Diberikan sejumlah data yang dideskripsikan oleh sejumlah atribut dan sebuah *class* hasil, permasalahannya adalah mencari pohon keputusan minimum yang dapat mengklasifikasikan nilai *class* berdasarkan nilai atributnya

# Pembuatan Pohon Keputusan

- Kumpulkan sampel untuk membuat training set
- Tentukan *class* dari tiap sampel
- Pilih salah satu atribut untuk menjadi titik awal atau node *root* dari *tree*
- Bagi training set menjadi sejumlah tabel, masing-masing tabel memiliki sampel dengan nilai atribut terpilih yang sama
- Jika nilai *class* dapat terbagi/terpartisi, maka proses selesai. Jika tidak, pilih atribut baru dan bagi kembali training set berdasarkan nilai atribut baru tersebut

# Algoritma untuk Induksi Pohon Keputusan

- Algoritma dasar:
  - Tree dibangun dengan pendekatan *divide-and-conquer top-down* rekursif
  - Di awal, semua sampel training ada di root
  - Atribut bertipe kategori (jika ada nilai kontinyu, maka perlu didiskritkan terlebih dahulu)
  - Sampel dipartisi secara rekursif berdasarkan pada atribut-atribut yang terpilih
  - Pemilihan atribut dilakukan berdasarkan nilai heuristik atau perhitungan statistik (contoh: entropi informasi)
- Kondisi yang menghentikan proses partisi rekursif:
  - Semua sampel pada sebuah node berada pada class yang sama
  - Tidak ada lagi atribut yang dapat digunakan untuk melakukan partisi → majority voting dilakukan untuk mengklasifikasi node daun
  - Tidak ada lagi sampel yang dapat digunakan

# Contoh

Atribut

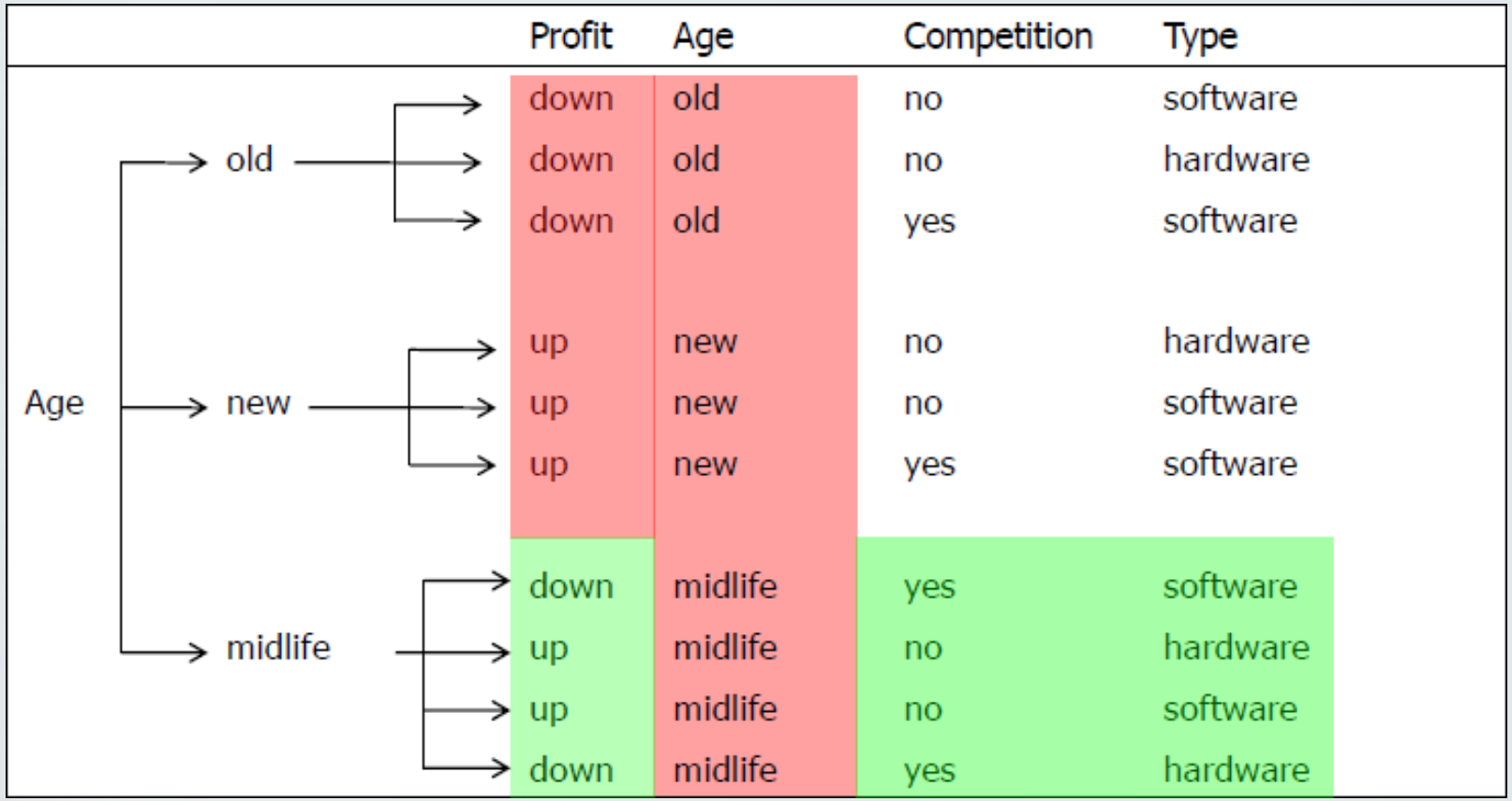
- Tentukan cara untuk memprediksi profil perusahaan yang akan menyebabkan kenaikan atau penurunan profit berdasarkan data berikut:

Class

| Profit | Age | Competition | Type |
|---|---|---|---|
| Down | Old | No | Software |
| Down | Midlife | Yes | Software |
| Up | Midlife | No | Hardware |
| Down | Old | No | Hardware |
| Up | New | No | Hardware |
| Up | New | No | Software |
| Up | Midlife | No | Software |
| Up | New | Yes | Software |
| Down | Midlife | Yes | Hardware |
| Down | Old | Yes | Software |

# Langkah Penyelesaian

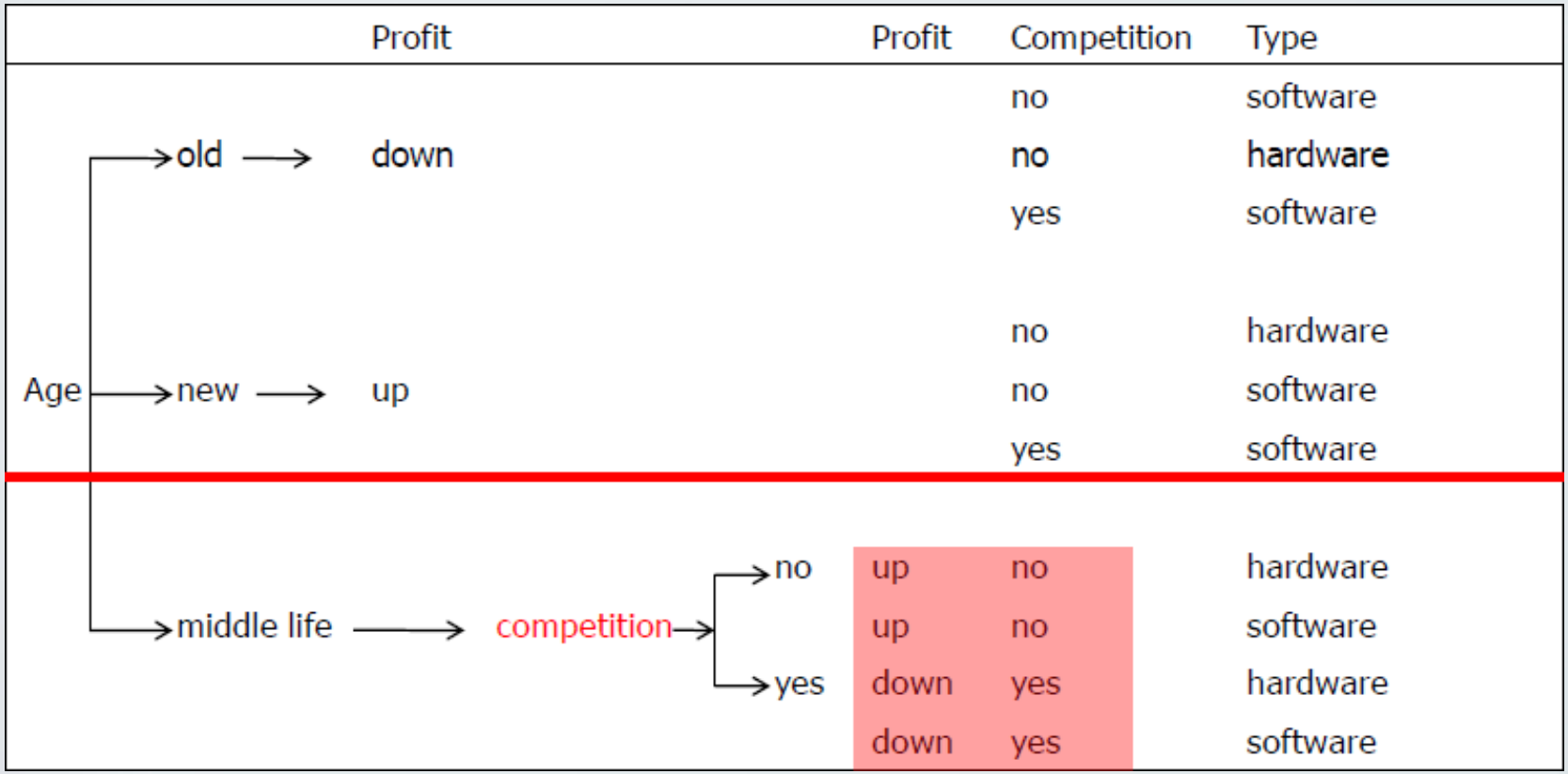
- Kita lakukan pemisahan berdasarkan atribut Age:




| Profit | Age | Competition | Type |
|---|---|---|---|
| down | old | no | software |
| down | old | no | hardware |
| down | old | yes | software |
| up | new | no | hardware |
| up | new | no | software |
| up | new | yes | software |
| down | midlife | yes | software |
| up | midlife | no | hardware |
| up | midlife | no | software |
| down | midlife | yes | hardware |

# Ukuran Training Set menjadi lebih kecil

- Setelah pemisahan ke-2 berdasarkan atribut Competition:



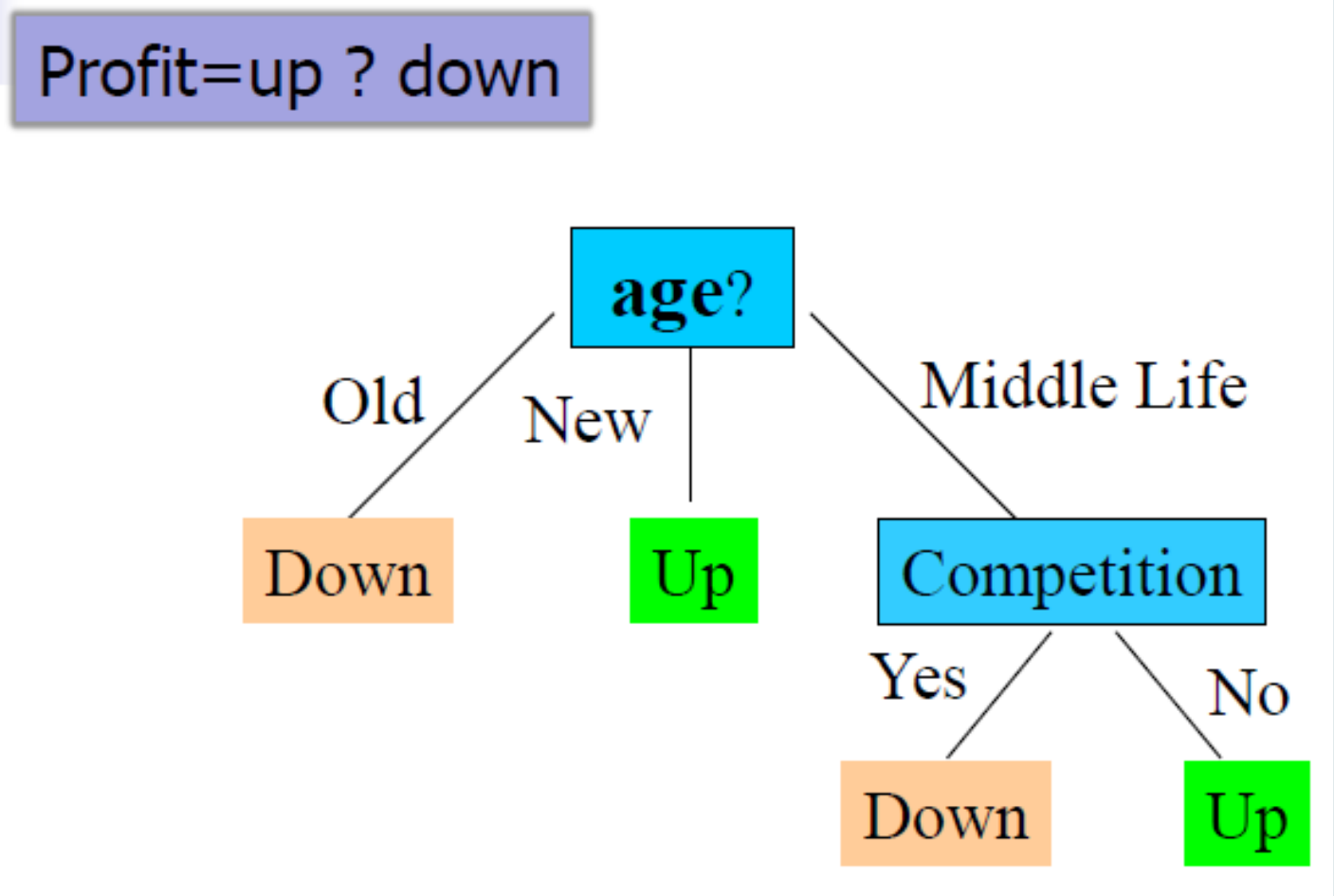
# Aturan yang didapat dari tree

| | IF | THEN |
|---|---|---|
| Rule1 | age is old | profit is down |
| Rule2 | age is new | profit is up |
| Rule3 | age is midlife AND competition is no | profit is up |
| Rule4 | age is midlife AND competition is yes | profit is down |

Dapatkah aturan ini diterapkan pada semua sampel training set?

Dapatkah aturan ini digunakan untuk memprediksi semua kasus pada permasalahan (memprediksi nilai class PROFIT)?

# Output: Pohon Keputusan untuk Profil Perusahaan



# Dimana harus memulai?

- Atribut mana dari AGE, TYPE, dan COMPETITION yang paling kuat berasosiasi dengan class PROFIT?
- Perhatikan bahwa:
  - PROFIT punya 2 state: UP atau DOWN
  - COMPETITION punya 2 state: NO atau YES
  - TYPE punya 2 state: SOFTWARE atau HARDWARE
  - AGE punya 3 state: OLD, NEW, atau MIDLIFE

# Kita perlu membedakan tingkat prediksi dari tiap atribut terhadap nilai class

| Competition | Profit |
|---|---|
| no | down |
| no | up |
| no | down |
| no | up |
| no | up |
| no | up |
| yes | down |
| yes | up |
| yes | down |
| yes | down |

Adakah indikasi hubungan antara nilai Competition dengan nilai Profit? → TIDAK

P(Profit = up | Comp = No) = 4/6 = .67
P(Profit = down | Comp = No) = 2/6 = .33
P(Profit = up | Comp = Yes) = 1/4 = .25
P(Profit = down | Comp = Yes) = 3/4 = .75

Harus dibandingkan dengan atribut lainnya!

# Menggunakan Probabilitas Kondisional untuk menunjukkan relasi antara atribut dan class

TYPE:

P( Profit= up | Type = Software ) = 0.5
P( Profit= down | Type = Software ) = 0.5
P( Profit= up | Type = Hardware ) = 0.5
P( Profit= down | Type = Hardware ) = 0.5

AGE:

P( Profit= up | Age = Old ) = 0.0
P( Profit= down | Age = Old ) = 1.0
P( Profit= up | Age = New ) = 1.0
P( Profit= down | Age = New ) = 0.0
P( Profit= up | Age = Midlife ) = 0.5
P( Profit= Down | Age = Midlife ) = 0.5

Jika Age=OLD, peluang Profit=UP adalah 0

# Permasalahan:
# Bagaimana cara memilih atribut?

- Kita memerlukan rumus untuk menentukan atribut yang paling penting/signifikan dalam perhitungan probabilitas atribut
- Diperlukan statistik prediktif berdasarkan probabilitas kondisional untuk memilih atribut “terbaik” pada setiap level

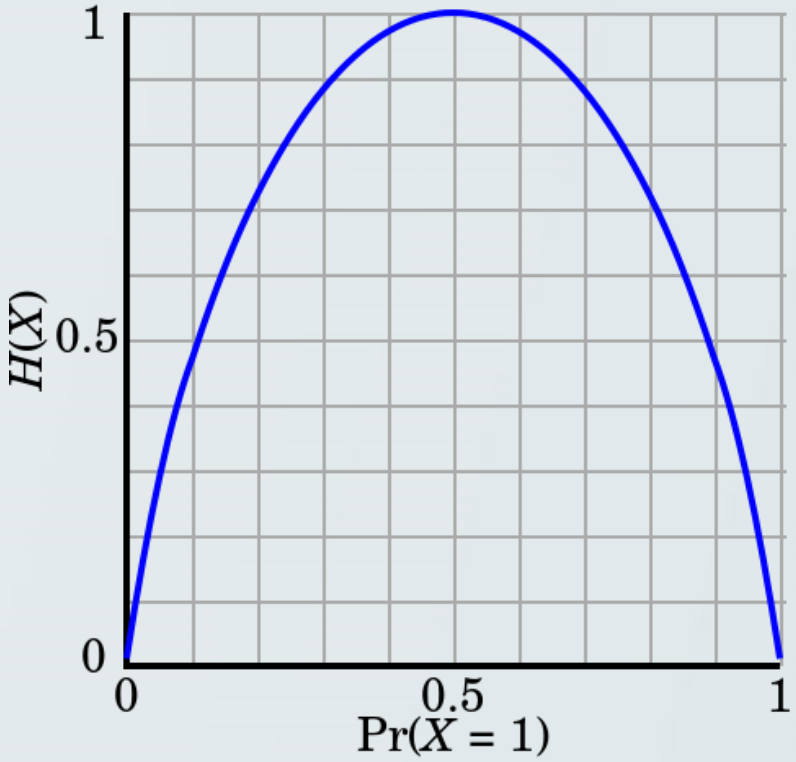
Di antara 3 atribut: AGE, COMPETITION, dan TYPE, mana yang membawa informasi “lebih banyak” dari yang lain?

# Pengukuran Informasi

- Mengapa informasi harus dapat diukur?
  - Teori Informasi
  - *Information retrieval,* proses *coding* (kompresi dan dekompresi)
- Apa kegunaan pengukuran informasi?
  - Penerapan pada data mining
  - Mengkuantifikasi faktor-faktor yang mempengaruhi pengambilan keputusan (untuk menentukan derajat kepentingan pesan yang kita terima)

# Teori Informasi

- Berapa banyak informasi yang dikandung oleh sebuah pesan bergantung pada seberapa tinggi tingkat ketidakpastiannya
- Semakin pasti sebuah pesan, semakin sedikit informasi yang dimiliki (dengan kata lain, jumlah informasi meningkat ketika probabilitas pesan menurun → berelasi secara terbalik)
- Semakin besar jumlah pesan yang mungkin, semakin besar jumlah informasi yang terkandung (sifat penambahan )



# Pengukuran Informasi

$$\log(\frac{x}{y}) = \log(x) - \log(y)$$
$$\log 1 = 0$$

- Informasi yang terkandung:

$$I(M) = \log(\frac{1}{p(M)}) = -\log(p(M))$$

  - Jika log2 digunakan, maka satuannya adalah bit
- Hukum keterbalikan antara Informasi dan Ketidakpastian
  - Pesan yang jarang muncul mengandung informasi lebih banyak daripada pesan yang sering muncul
  - Sebuah pesan yang pasti (memiliki probabilitas 1), memiliki 0 informasi

Berapa jumlah informasi dari serangkaian pesan yang masing2nya memiliki probabilitas yang berbeda?

- Rata-rata informasi yang terkandung dihitung dengan jumlah dari probabilitas dikalikan dengan bit informasi dari tiap pesan
- Contoh: Dalam sebuah pesan yang berisi 2 huruf A dan B dengan probabilitas 2/3 dan 1/3, rata-rata informasi yang terkandung:

$$\frac{2}{3} * log_2\left(\frac{1}{2/3}\right) + \frac{1}{3} * log_2\left(\frac{1}{1/3}\right) = 0.918$$

# Rumus Shannon

- Entropi Informasi
  - Pengukuran derajat ketidakpastian. Didefinisikan sebagai rata-rata informasi terkandung dari sebuah pesan *m* dari keseluruhan pesan
  - Pengukuran ketidakpastian klasifikasi dari sebuah objek terhadap keseluruhan objek yang diklasifikasikan
- Diberikan sekumpulan objek C dan partisi $c_1$, …, $c_n$, entropi dari klasifikasi ini:

$$H(C) = -\sum_{i=1}^{n} p(c_i) * log_2(p(c_i))$$

dimana $P(C_i)$ adalah probabilitas dari partisi $c_i$.

# Contoh: Informasi terkandung untuk class PROFIT

$$H(C)$$
$$= -[(p(Profit = up) * log_2(p(Profit = up))$$
$$+ p(Profit = down) * log_2(p(Profit = down))]$$
$$H(C) = -[0{,}5 * (-1) + 0{,}5 * (-1)]$$
$$H(C) = 1$$

- Ini menunjukkan bahwa 1 bit informasi dibutuhkan untuk mewakili 2 state berbeda dari PROFIT: up dan down. Belum ada informasi bagaimana nilai up dan down diklasifikasikan berdasarkan nilai dari atribut-atribut yang lain

# Penggunaan Rumus Shannon untuk menentukan atribut yang paling signifikan

- Untuk probabilitas sebuah nilai atribut (mis. AGE=old), berapa entropi informasi dari class C?

$$H(C \mid a_j) = -\sum_{i=1}^{n} p(c_i \mid a_j) \times \log_2(p(c_i \mid a_j)) \quad \text{(Formula 1)}$$

- Dimana i=1..n (sebanyak nilai class). Fungsi $p(c_i|a_j)$ adalah probabilitas nilai class adalah $c_i$ ketika atribut memiliki nilai j

# AGE=old

$$H(C \mid a_j) = -\sum_{i=1}^{n} p(c_i \mid a_j) \times \log_2 (p(c_i \mid a_j))$$ (Formula 1)

H(PROFIT|Age=old)

= -p(PROFIT=up|Age=old)*$\log_2$(p(PROFIT=up|Age=old))

  -p(PROFIT=down|Age=old)*$\log_2$(p(PROFIT=down|Age=old))

= -0 * $\log_2$ 0 – 1 * $\log_2$ 1

= 0

# Quiz

- Bagaimana nilai untuk H(Profit|Age=new) dan H(Profit|Age=Middlelife)?

$$H(C \mid a_j) = -\sum_{i=1}^{n} p(c_i \mid a_j) \times \log_2(p(c_i \mid a_j))$$ (Formula 1)

# Age=new/middlelife

$$H(C \mid a_j) = -\sum_{i=1}^{n} p(c_i \mid a_j) \times \log_2(p(c_i \mid a_j))$$

$$\begin{aligned}
&H(\text{PROFIT} \mid \text{Age} = \text{new}) \\
&= -p(\text{PROFIT} = \text{up} \mid \text{Age} = \text{new}) \times \log_2(p(\text{PROFIT} = \text{up} \mid \text{Age} = \text{new})) \\
&\quad - p(\text{PROFIT} = \text{down} \mid \text{Age} = \text{new}) \times \log_2(p(\text{PROFIT} = \text{down} \mid \text{Age} = \text{new})) \\
&= -1 \times \log_2 1 - 0 \times \log_2 0 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
&H(\text{PROFIT} \mid \text{Age} = \text{midlife}) \\
&= -p(\text{PROFIT} = \text{up} \mid \text{Age} = \text{midlife}) \times \log_2(p(\text{PROFIT} = \text{up} \mid \text{Age} = \text{midlife})) \\
&\quad - p(\text{PROFIT} = \text{down} \mid \text{Age} = \text{midlife}) \times \log_2(p(\text{PROFIT} = \text{down} \mid \text{Age} = \text{midlife})) \\
&= -0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 \\
&= 1
\end{aligned}$$

# Bagaimana setiap atribut berkontribusi terhadap pengklasifikasian *class*?

$$H(C \mid A) = \sum_{j=1}^{m} [p(a_j) \times H(C \mid a_j)]$$ (Formula 2)

- Dimana j=1..m. m adalah total jumlah nilai dari atribut A

$$\begin{aligned}
&H(\text{PROFIT} \mid \text{Age}) \\
&= p(\text{Age} = \text{new}) \times H(\text{PROFIT} \mid \text{Age} = \text{new}) + p(\text{Age} = \text{old}) \times H(\text{PROFIT} \mid \text{Age} = \text{old}) \\
&+ p(\text{Age} = \text{midlife}) \times H(\text{PROFIT} \mid \text{Age} = \text{midlife}) \\
&= \frac{3}{10} \times 0 + \frac{3}{10} \times 0 + \frac{4}{10} \times 1 \\
&= 0.4
\end{aligned}$$

# Quiz

- Bagaimana nilai untuk H(Profit|Competition) dan H(Profit|Type)?
  - Ingat:
    - Competition punya 2 nilai: YES dan NO
      - H(Profit|Competition=Yes)
      - H(Profit|Competition=No)
    - Type punya 2 nilai: SOFTWARE dan HARDWARE
      - H(Profit|Type=Software)
      - H(Profit|Type=Hardware)

# H(Profit|Competition) dan H(Profit|Type)

$$H(C \mid A) = \sum_{j=1}^{m} [p(a_j) \times H(C \mid a_j)]$$

$$H(\text{PROFIT} \mid \text{Comp})$$
$$= p(\text{Comp} = \text{yes}) \times H(\text{PROFIT} \mid \text{Comp} = \text{yes}) + p(\text{Comp} = \text{no}) \times H(\text{PROFIT} \mid \text{Comp} = \text{no})$$
$$= p(\text{Comp} = \text{yes}) \times (H(\text{PROFIT} = \text{up} \mid \text{Comp} = \text{yes}) + H(\text{PROFIT} = \text{down} \mid \text{Comp} = \text{yes}))$$
$$+ p(\text{Comp} = \text{no}) \times (H(\text{PROFIT} = \text{up} \mid \text{Comp} = \text{no}) + H(\text{PROFIT} = \text{down} \mid \text{Comp} = \text{no}))$$
$$= \frac{4}{10} \times (-\frac{1}{4} \times \log_2(\frac{1}{4}) - \frac{3}{4} \times \log_2(\frac{3}{4})) + \frac{6}{10} \times (-\frac{4}{6} \times \log_2(\frac{4}{6}) - \frac{2}{6} \times \log_2(\frac{2}{6}))$$
$$= \frac{4}{10} \times 0.81 + \frac{6}{10} \times 0.92 = 0.88$$

$$H(\text{PROFIT} \mid \text{Type})$$
$$= p(\text{Type} = \text{software}) \times H(\text{PROFIT} \mid \text{Type} = \text{software}) + p(\text{Type} = \text{software}) \times H(\text{PROFIT} \mid \text{Type} = \text{hardware})$$
$$= \frac{6}{10} \times (-\frac{1}{2} \times \log_2(\frac{1}{2}) - \frac{1}{2} \times \log_2(\frac{1}{2})) + \frac{4}{10} \times (-\frac{1}{2} \times \log_2(\frac{1}{2}) - \frac{1}{2} \times \log_2(\frac{1}{2}))$$
$$= 1$$

# Bagaimana menentukan atribut yang signifikan?

- Pilih atribut yang memiliki entropi terkecil
  - H(Profit|Age)=0,4
  - H(Profit|Competition)=0,8
  - H(Profit|Type)=1

Age adalah atribut terbaik

$$\min_{t=1}^{n}\{H(C|A_t\}$$ (Formula 3)

Dimana t=1..n. n adalah total jumlah atribut untuk class C

# Proses Induksi Menggunakan ID3

- Menentukan atribut utama dan output
- Membuat training set
- Mengkonversi data numerik ke nilai diskrit dengan menggunakan range
- Terapkan ID3 secara rekursif:
  - Gunakan ketiga formula untuk memilih atribut untuk memisahkan training set (urut berdasarkan atribut dan pisahkan baris yang nilai class-nya cocok dengan nilai atribut)
  - Ulangi proses di atas dengan menggunakan ketiga formula untuk memilih atribut lain untuk memisahkan lebih jauh lagi training set
- Lakukan proses pemangkasan *(pruning)* pada tree

# Pseudocode pembentukan tree pada ID3

```
   Input  : D, attribute_list;
   Output: A decision tree.
 1 create a node N;
 2 if samples in D are all of the same class, C then
 3 |   return N as a leaf node labeled with C;
 4 end
 5 if attribute_list is empty then
 6 |   return N as a leaf node leaf node labeled with the majority
   |   class in D;
 7 end
 8 apply Attribute_selection_method (D,attribute_list) to find
   the "best" attribute A*;
 9 label node N with A*;
10 attribute_list ← attribute_list − A*;
11 foreach value j of attribute A* do
12 |   let Dj be the set of samples in D satisfying value j;
13 |   if Dj is empty then
14 |   |   label N with the majority class in D;
15 |   end
16 |   else
17 |   |   label N with the node returned by
   |   |   Generate_decision_tree (Dj, attribute_list);
18 |   end
19 end
20 return N;
```

# Permasalahan pada ID3

- Noisy data: pengukuran yang salah, error saat memasukkan data
- Nilai atribut yang kosong
- Percabangan yang terlalu banyak
- ID3 disempurnakan lagi dengan algoritma C4.5
- C4.5 disempurnakan kembali dalam C5

# Tugas

- Lakukan algoritma ID3 untuk menghasilkan pohon keputusan dan aturan yang digunakan untuk pembelian komputer seperti data berikut:

| Age | Student | Credit_Rating | Buys_Computer |
|---|---|---|---|
| Middle_Aged | No | Excellent | Yes |
| Youth | No | Fair | No |
| Senior | No | Fair | No |
| Youth | Yes | Excellent | Yes |
| Middle_Aged | No | Fair | No |
| Senior | Yes | Fair | Yes |
| Youth | No | Excellent | No |
| Senior | Yes | Excellent | Yes |

# Tips Pengerjaan

- Hitung dulu probabilitas kondisional dari setiap atribut

| | Age=Middle_Aged | Age=Youth | Age=Senior |
|---|---|---|---|
| Buys_Computer=Yes | | | |
| Buys_Computer=No | | | |

….. Lakukan untuk kedua atribut yang lain

- Hitung entropi dari seluruh atribut

| | Entropy | Buys_Computer = Yes | Buys_Computer = No |
|---|---|---|---|
| H(C\|Age=Middle_Aged)<br>H(C\|Age=Youth)<br>H(C\|Age=Senior)<br><br>H(C\|Age) | | | |
| H(C\|Student=Yes)<br>H(C\|Student=No)<br><br>H(C\|Student) | | | |
| H(C\|Credit_Rating=Excellent)<br>H(C\|Credit_Rating=Fair)<br><br>H(C\|Credit_Rating) | | | |

# Tips Pengerjaan

- Ambil atribut dengan entropi terkecil dan bagi data tabel sesuai dengan nilai atribut tersebut
- Buang data tabel yang sudah memiliki nilai class yang sama dengan pembagian atribut
- Ulangi proses pada sisa data tabel yang belum memiliki class yang sama untuk tiap atributnya
- Buat pohon ID3-nya
- Buat tabel aturannya