Session 31

Classification I

Table of Content

Apa yang Akan Kita Pelajari Hari Ini?

1. KNN
2. Decision Tree
3. Ensemble Method

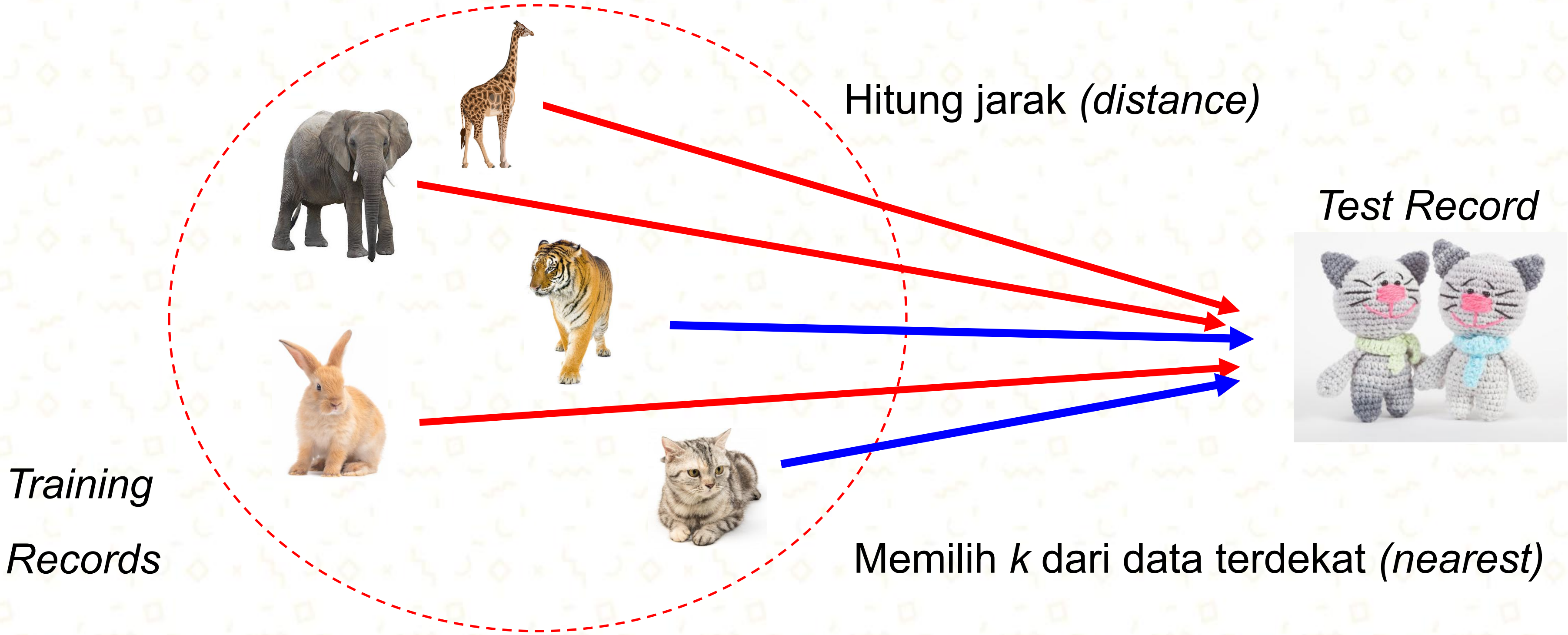
KNN

Classification and Clustering Techniques

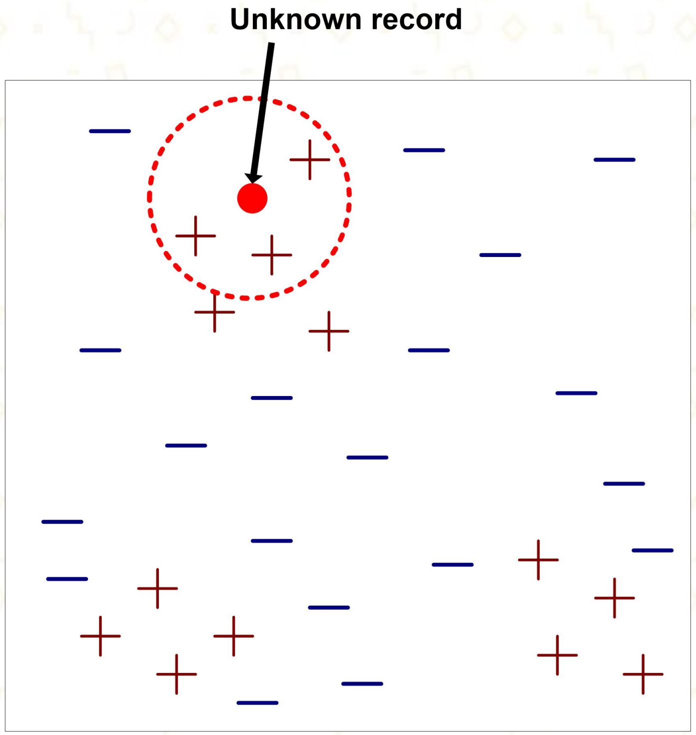
- Classification
  - K-Nearest Neighbour
  - Decision Tree
  - Ensemble Methods
- Clustering
  - K-medoid
  - K-means
  - DBSCAN

Nearest Neighbor Classifiers

- Ide dasar:
  - Jika dia berjalan seperti kucing, mengeong seperti kucing, maka itu mungkin kucing

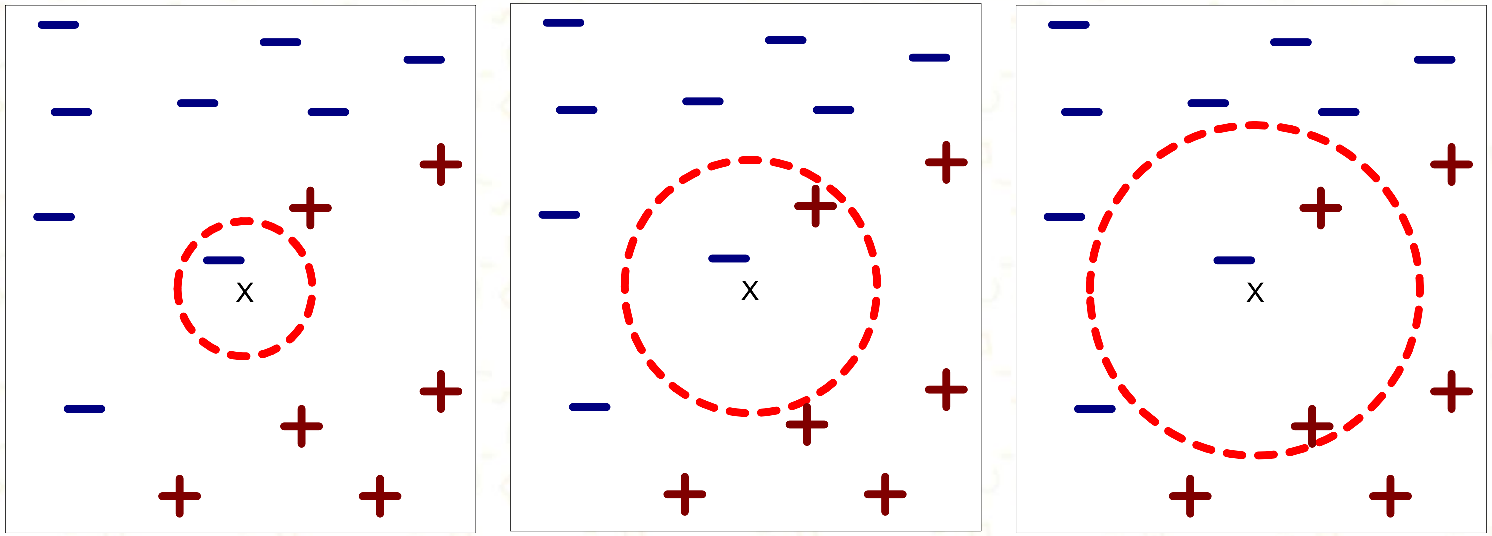


Nearest Neighbor Classifiers



- **Membutuhkan tiga hal**
  - Kumpulan data yang tersimpan
  - Jarak Metrik *(distance metric)* untuk menghitung jarak antar data
  - Nilai *k*, jumlah tetangga terdekat *(nearest neighbour)* yang akan diambil
- **Untuk mengklasifikasi data baru:**
  - Hitung jarak terhadap data lain
  - Memilih *k* tetangga terdekat (nearest neighbour)
  - Gunakan label kelas dari tetangga terdekat untuk menentukan label kelas dari data baru (misalnya, dengan mengambil suara mayoritas)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- *K-nearest neighbor* dari record x adalah
  - *k* data yang memiliki jarak terdekat ke x

Nearest Neighbor Classification

- Hitung jarak antara dua titik:
  - *Euclidean distance*

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Memilih class dari tetangga terdekat
  - Ambil suara mayoritas dari label kelas di antara *k-nearest neighbour*
  - Memberi bobot suara menurut jarak *(distance)*

$$w_i = \frac{1}{d(x, x_i)} \qquad W = \sum w_i \qquad y = \sum_{i=1}^{k} \frac{w_i}{W} y_i$$

k-NN - Example

- *Given 14 examples → map to 4-D space*
- *Classify unknown sample*

X: (age="<30", income="medium", student="yes", credit="fair")
→ (0, 0.5, 1, 0)

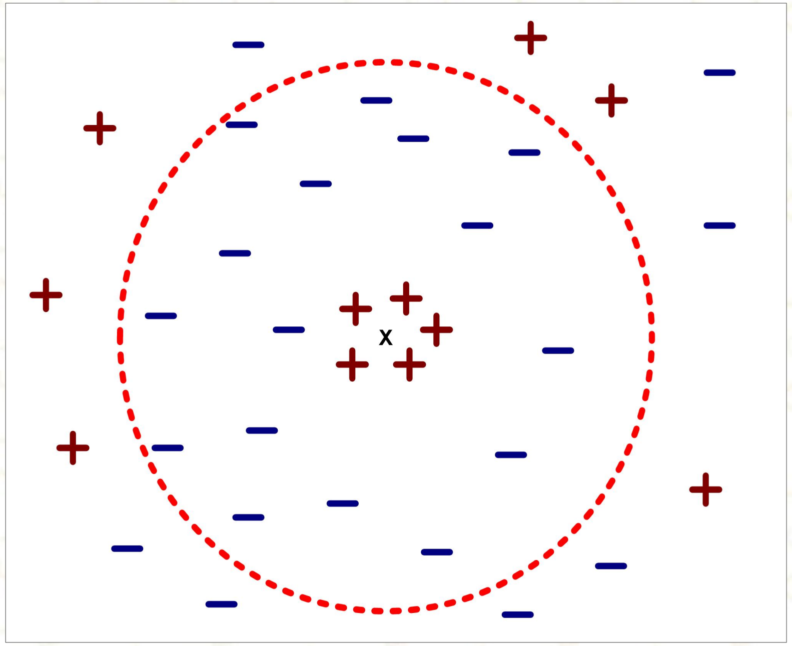
- 3-NN: **(0, 0, 1, 0): 1(yes),** d = 0.5, w = 1/0.5
  (0, 0.5, 0, 0): 0(no), d = 1.0, w = 1/1.0
  (0, 0.5, 1, 1): 1(yes), d = 1.0, w = 1/1.0
- W = 4
- y = 2/4 x 1(yes) + 1/4 x 0(no) + 1/4 x 1(yes) = 0.75

→ Classify X as "**Yes**"

No.	age	income	student	credit_rating	buys_computer
1	0	1	0	0	0
2	0	1	0	1	0
3	0.5	1	0	0	1
4	1	0.5	0	0	1
5	1	0	1	0	1
6	1	0	1	1	0
7	0.5	0	1	1	1
8	0	0.5	0	0	0
9	0	0	1	0	1
10	1	0.5	1	0	1
11	0	0.5	1	1	1
12	0.5	0.5	0	1	1
13	0.5	1	1	0	1
14	1	0.5	0	1	0

Nearest Neighbor Classification

- Choosing the value of *k:*
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes



Nearest Neighbor Classification

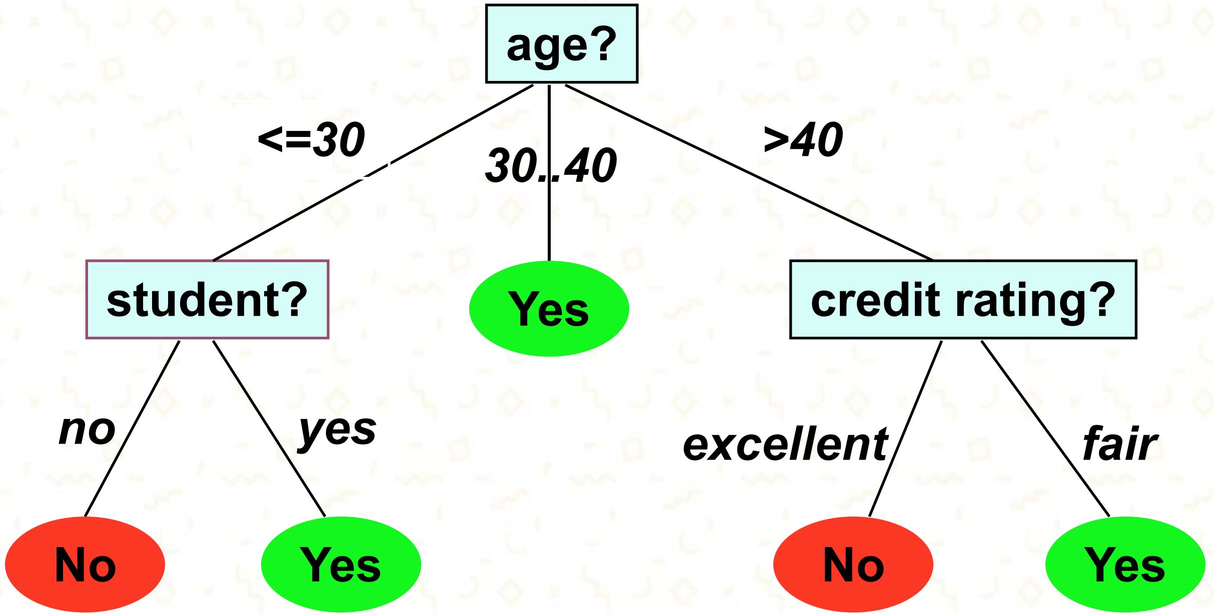
- Permasalahan *Scaling*
  - Attributes perlu di-*scale*
    - Untuk mencegah jarak *(distance)* didominasi oleh salah satu atribut
    - Contoh:
    - tinggi seseorang dapat bervariasi dari 1,5 m hingga 1,8 m
    - berat seseorang dapat bervariasi dari 40kg hingga 150kg
    - pendapatan seseorang dapat bervariasi dari $10K hingga $1M

Decision Tree

Training Dataset

No.	age	income	student	credit_rating	buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31…40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31…40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Output: A Decision Tree for buys_computer

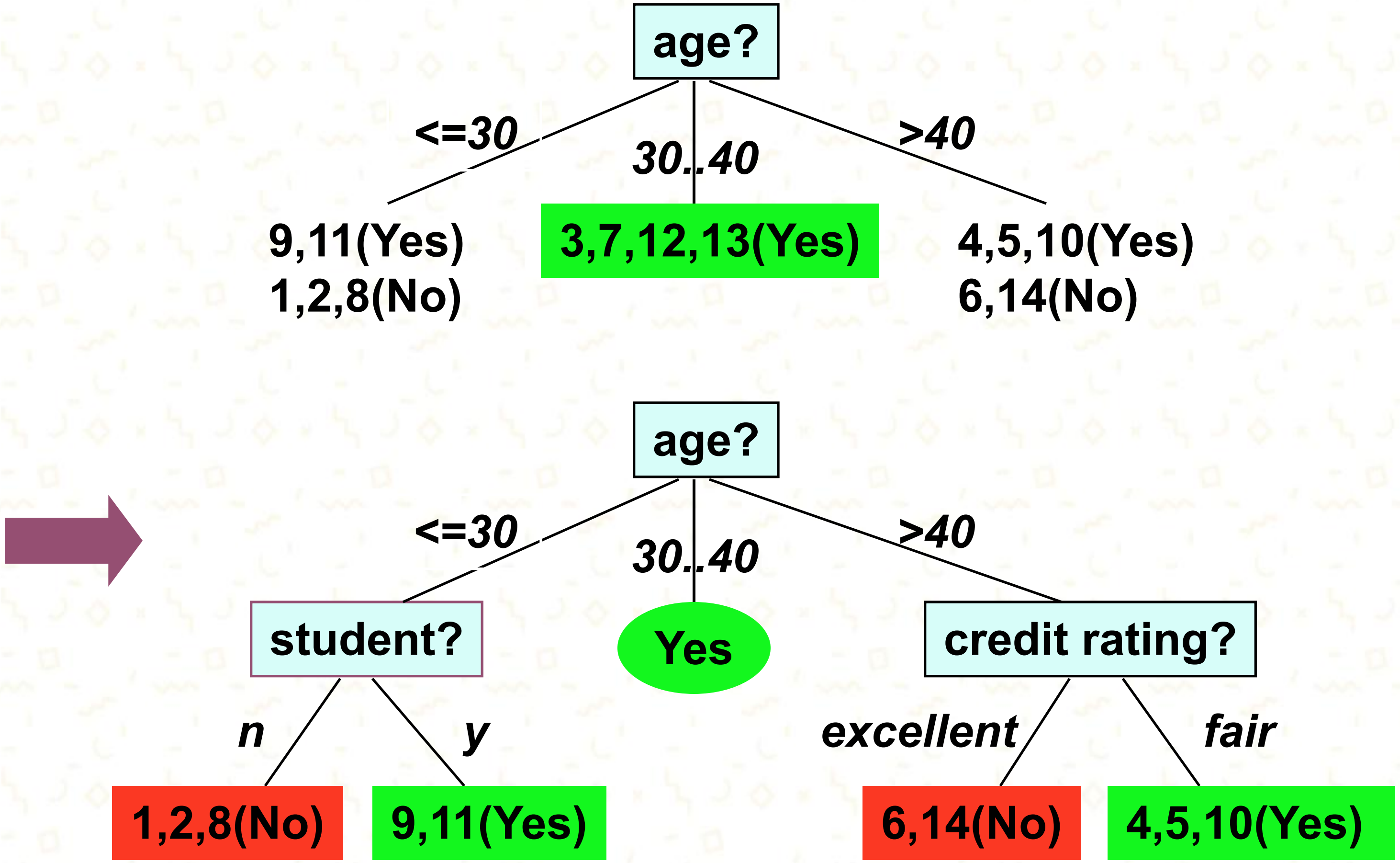


X: (age="<30", income="medium", student="yes", credit="fair") ⇒ Yes

Algorithm for DT Induction

- *Top-down recursive divide-and-conquer manner*
  - Pada awalnya, semua training data ada di *root*
  - Attribute yang di *node*, dipilih berdasarkan heuristik atau statistik (mis., *information gain*)
  - Training data dipartisi secara rekursif berdasarkan atribut yang dipilih
- Kondisi untuk menghentikan partisi
  - Semua sampel termasuk dalam kelas yang sama
  - Tidak ada sampel yang tersisa
  - Tidak ada atribut yang tersisa

Algorithm for DT Induction



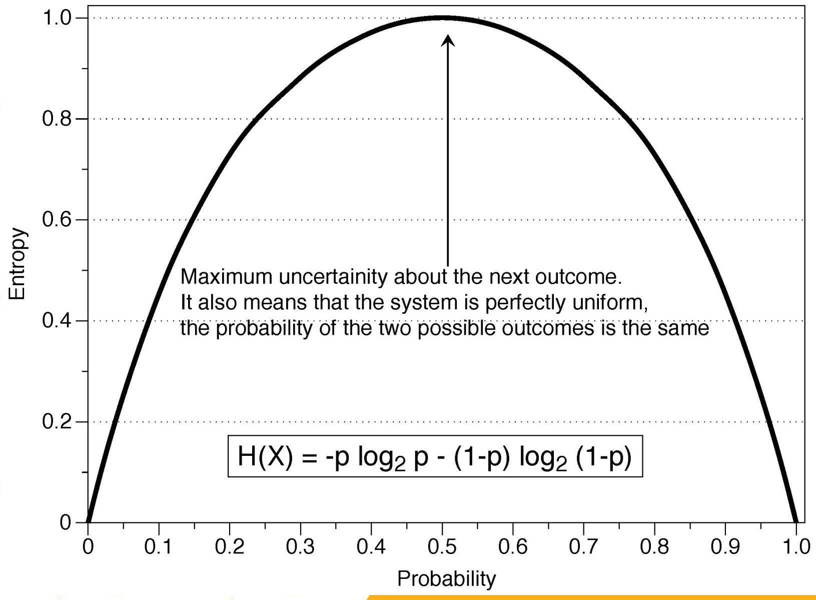
Entropy

- Entropi dapat didefinisikan sebagai ukuran kemurnian *sub split.*

$$E(S) = \sum_{i=1}^{m} p_i(-\log_2 p_i) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

Examples

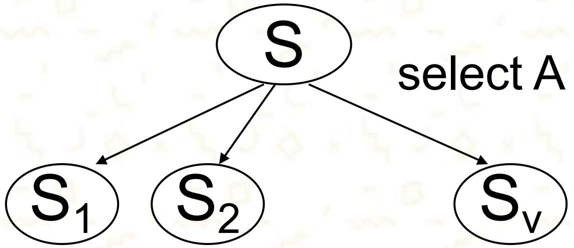
- S: {Y(1,2), N(3,4)} for instances 1,2,3,4
  → E(S) = (-1/2)*log(1/2) + (-1/2)*log(1/2) = 1
- S: {Y(1,3,4), N(2)} for instances 1,2,3,4
  → E(S) = (-3/4)*log(3/4) + (-1/4)*log(1/4) = 0.81
- S: {Y(1,2,3,4), N()} for instances 1,2,3,4
  → E(S) = (-4/4)*log(4/4) + (-0/4)*log(0/4) = 0



Information Gain (ID3/C4.5)

- Entropi yang diharapkan setelah memeriksa nilai atribut A
  = Rata rata entropy dari S_1, S_2, … S_n setelah partisi S menggunakan atribut A dengan nilai $\{a_1, a_2, \ldots, a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_j}{s} E(s_j) = \frac{s_1}{s} E(s_1) + \frac{s_2}{s} E(s_2) + \ldots + \frac{s_v}{s} E(s_v)$$



- Hitung *information gain* dari *attribute A*

$$Gain(A) = E(S) - E(A)$$

- Pilih atribut dengan *Information gain* yang terbesar

Attribute Selection by IG - Example

- C_1: buys_computer = “yes”, C_2: buys_computer = “no”
- **E(D)** = E(9,5) = $$-\frac{9}{14}\log\frac{9}{14}-\frac{5}{14}\log\frac{5}{14}=0.94$$

$$\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)$$

age	C1	C2	$E(S_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

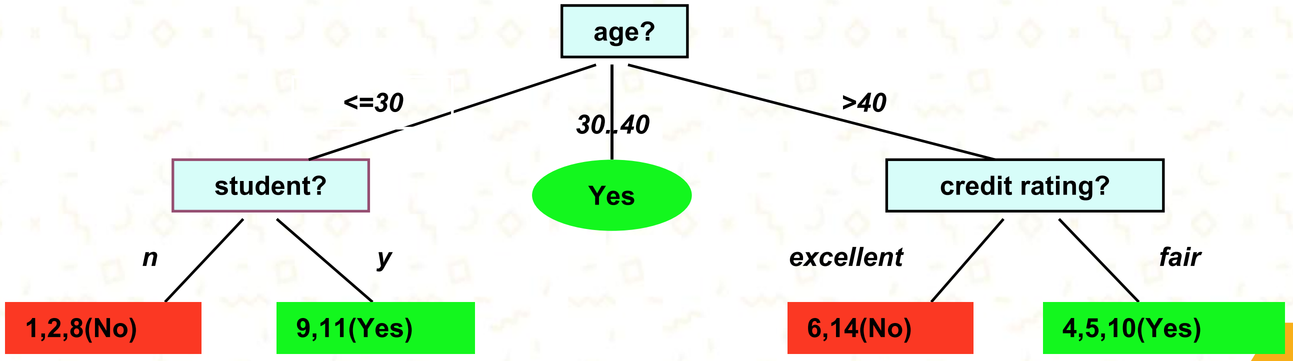
- **E(age)** = $$\frac{5}{14}E(\text{"}<=30\text{"})+\frac{4}{14}E(\text{"}30..40\text{"})+\frac{5}{14}E(\text{"}>40\text{"})=0.69$$
- **Gain(age)** = $$E(D)-E(age)=0.25$$

Gain(income) = 0.03, Gain(student) = 0.15

Extracting Classification Rules

- Merepresentasikan pengetahuan dalam bentuk *IF-THEN rules*
  - Satu aturan (rule) dibuat untuk setiap jalur dari akar ke daun
  - Node daun menyimpan prediksi kelas
- Rules mudah dipahami manusia
- Example

IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"
IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"
IF *age* = "31…40" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit* = "*excellent*" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit* = "*fair*" THEN *buys_computer* = "*no*"



Avoid Overfitting

- Tree yang dibuat mungkin akan *overfit* terhadap training data
  - Terlalu banyak cabang, diakibatkan oleh ouliers data
  - Hasilnya akurasi yang rendah terhadap data testing

- *Prepruning*
  - Hentikan konstruksi pohon lebih awal—jangan membagi simpul jika ini akan mengakibatkan akurasi jatuh di bawah ambang batas
- *Postpruning*
  - Hapus cabang dari Tree / pohon yang telalu lebat.
  - Jika memangkas / *pruning* sebuah node menghasilkan tingkat error yang lebih kecil (terhadap test set), silahkan di *pruning*

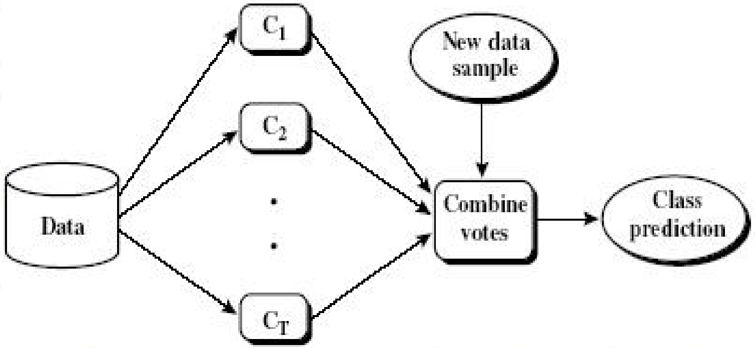
Discussion on DT

- Kelebihan
  - Aturan klasifikasi yang dapat dipahami oleh manusia
  - Kecepatan belajar/klasifikasi yang relatif lebih cepat
- Kekurangan
  - Sensitive (not robust/ tidak kuat) terhadap noises
  - Atribut bernilai kontinu - secara dinamis mempartisi nilai atribut kontinu ke dalam set interval diskrit

Ensemble Methods

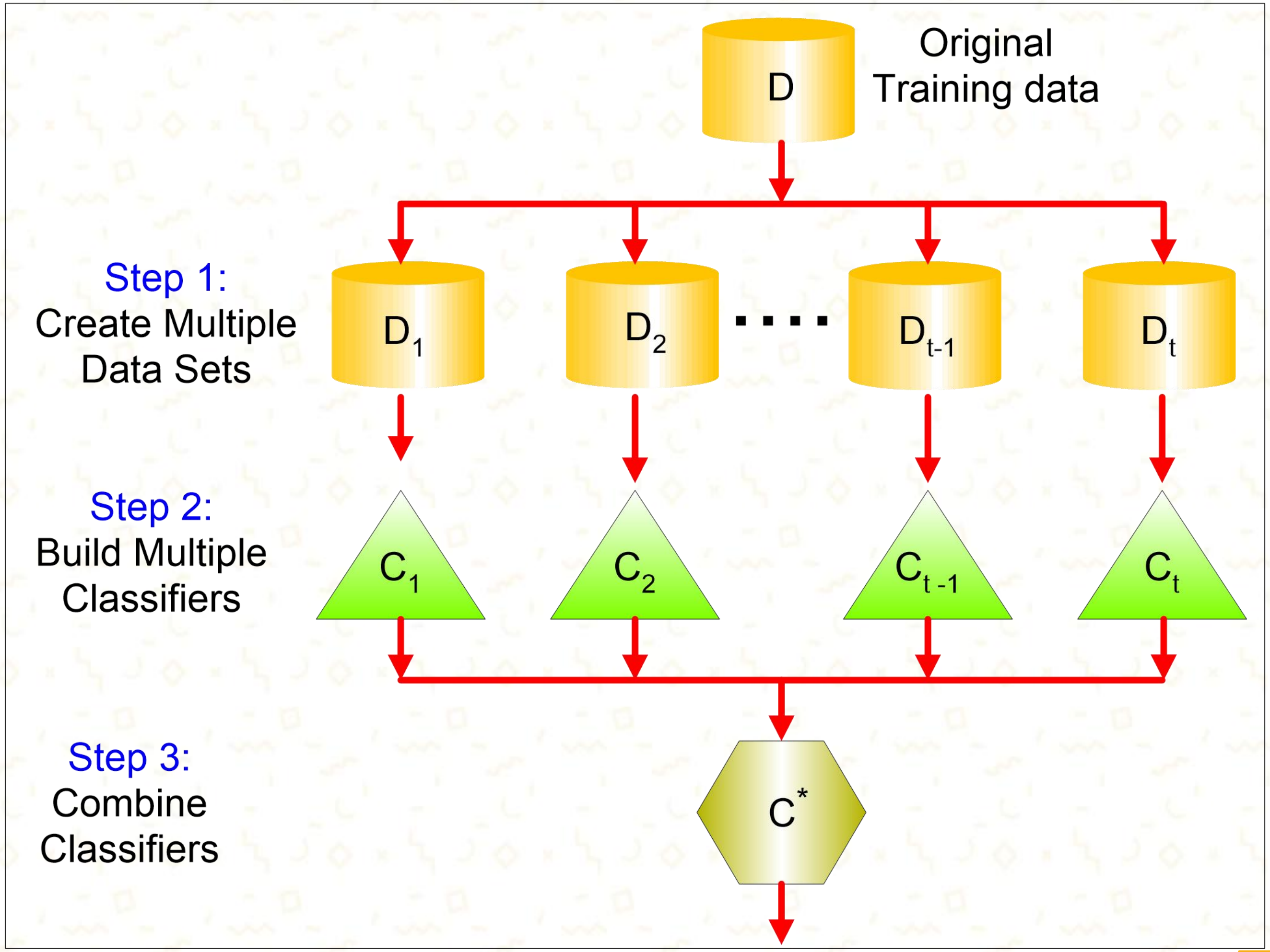
Ensemble Methods

- Ensemble
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_k, with the aim of creating an improved model M^*



- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers

General Idea



Bagging

- Analogy
  - Diagnosis based on multiple doctors' majority vote
- Training
  - Given a set *D* of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is sampled with replacement from *D* (i.e., boostrap)
  - A classifier model M_i is learned for each training set D_i
- Classification
  - Each classifier M_i returns its class prediction
  - The bagged classifier *M** counts the votes and assigns the class with the most votes to an unknown sample *X*
- Accuracy
  - Often significant better than a single classifier derived from *D*
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction

Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

c1
c2
c3
c4

- Build classifier on each bootstrap sample
- Counts the votes and assigns the class with the most votes to an unknown sample *X*

- Example : *Random Forest*

Boosting

- Analogy
  - Consult several doctors, based on a combination of diagnoses
  - Weight assigned based on the previous diagnosis accuracy
- Training
  - Weights are assigned to each training tuple
  - A series of *k* classifiers is iteratively learned
  - After a classifier M_i is learned, the weights are updated to allow the subsequent classifier, M_{i+1}, to pay more attention to the training tuples that were misclassified by M_i
- Classification
  - The final M^* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy
- Accuracy
  - Comparing with bagging: boosting tends to achieve greater accuracy, but it also risks overfitting the model to misclassified data

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

c1
c2
c3
c4

- Example 4 is hard to classify
- Its weight is increased; therefore it is more likely to be chosen again in subsequent rounds

- Example : *AdaBoost*

Lets Practice!

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU