

Pohon Klasifikasi

Bagus Sartono (Departemen Statistika – IPB)

Nama Lain

- Classification Tree
- Decision Tree
- Recursive Partition
- Iterative Dichotomiser

Kegunaan

- Mengidentifikasi variabel apa yang dapat dijadikan sebagai pembeda antar kelompok
- Memprediksi keanggotaan kelompok suatu individu berdasarkan karakteristiknya
- Terapannya antara lain:
 - Marketing: Mengidentifikasi prospective customer (cross-sell, up-sell, new acquisition)
 - Risk: Credit scoring, menentukan apakah calon penerima kredit akan mampu bayar atau tidak
 - Customer Relationship: churn analysis, menentukan customer yang berpotensi akan meninggalkan jasa/produk
 - Health: menentukan tingkat resiko penyakit
 - dll

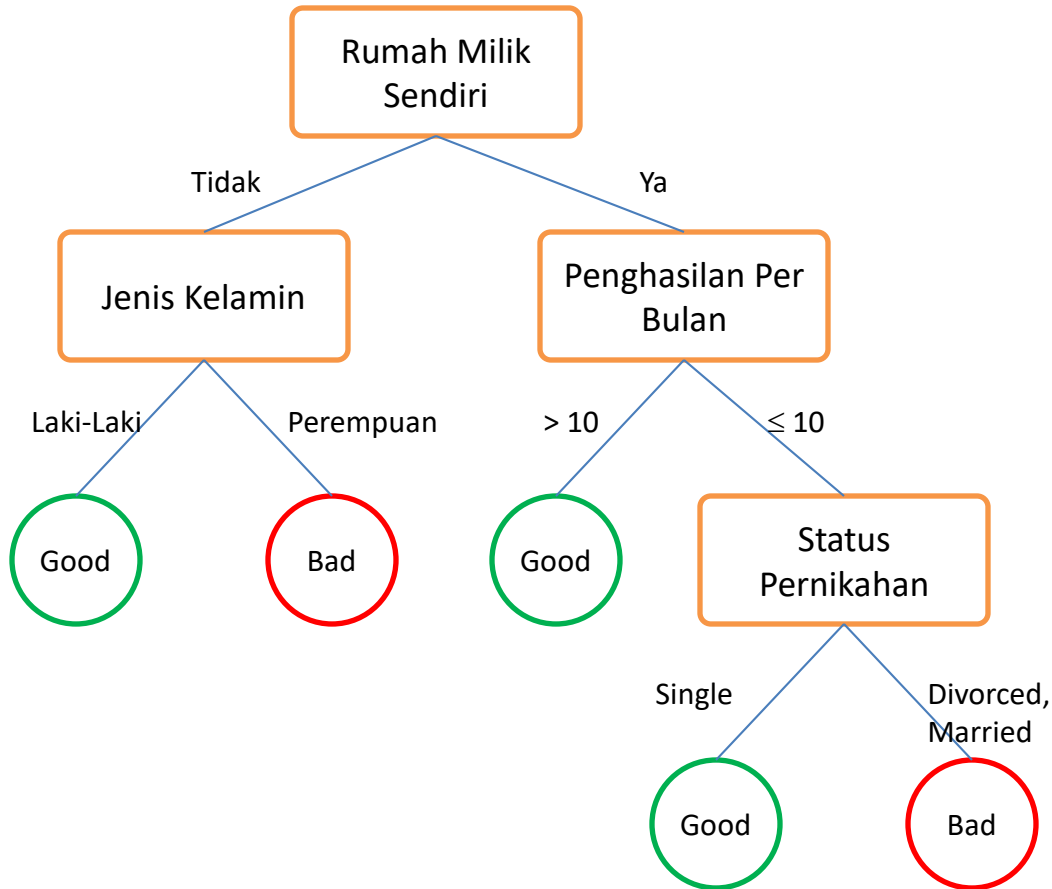
Metode lain yang setara kegunaannya

- Regresi Logistik
- Discriminant Analysis
- Support Vector Machine
- Bayesian Classifier
- Neural Network
- dll

Outline

- Pengenalan Konsep Entropy dan Information Gain
- Pengenalan Algoritma Dasar Pohon Klasifikasi
- Menilai Kemampuan Prediksi Pohon Klasifikasi
- Pengembangan Lebih Lanjut dari Pohon Klasifikasi

Ilustrasi penggunaan pohon klasifikasi

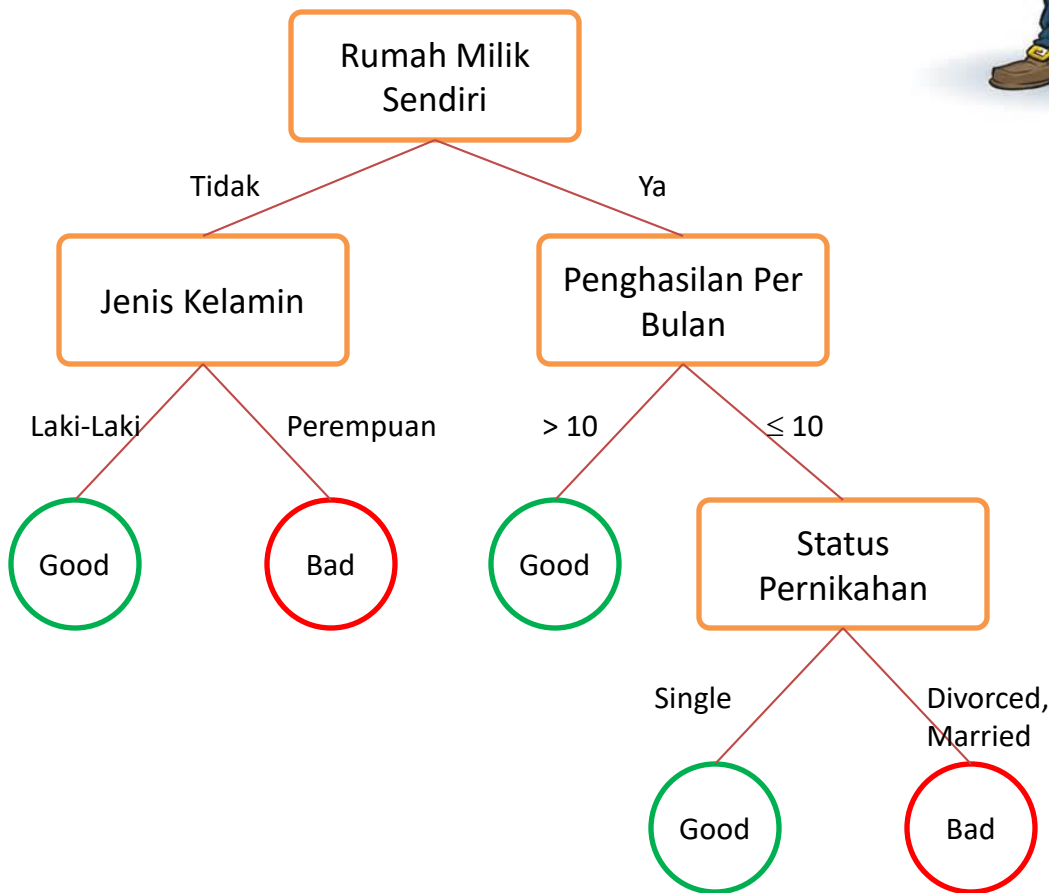


Ilustrasi penggunaan pohon klasifikasi



Profil:

- Pria
- Rumah Sendiri
- Penghasilan 8 juta per bulan
- Bujangan

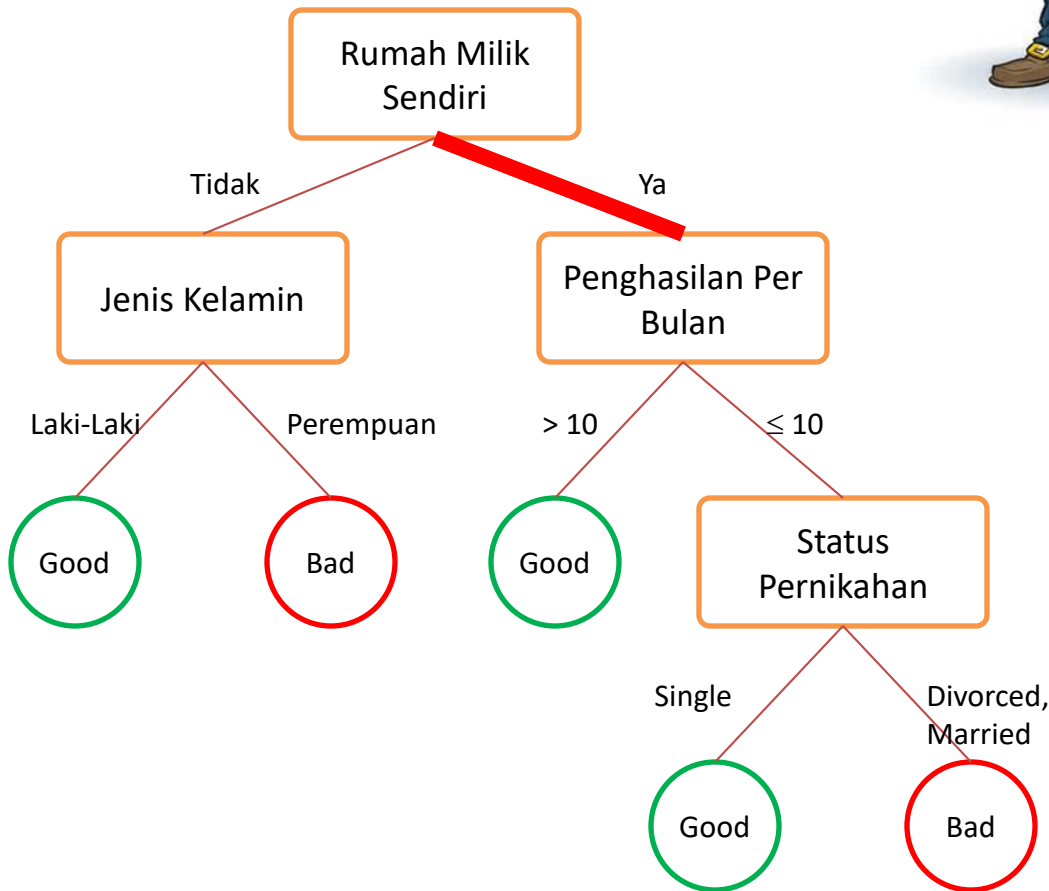


Ilustrasi penggunaan pohon klasifikasi



Profil:

- Pria
- **Rumah Sendiri**
- Penghasilan 8 juta per bulan
- Bujangan

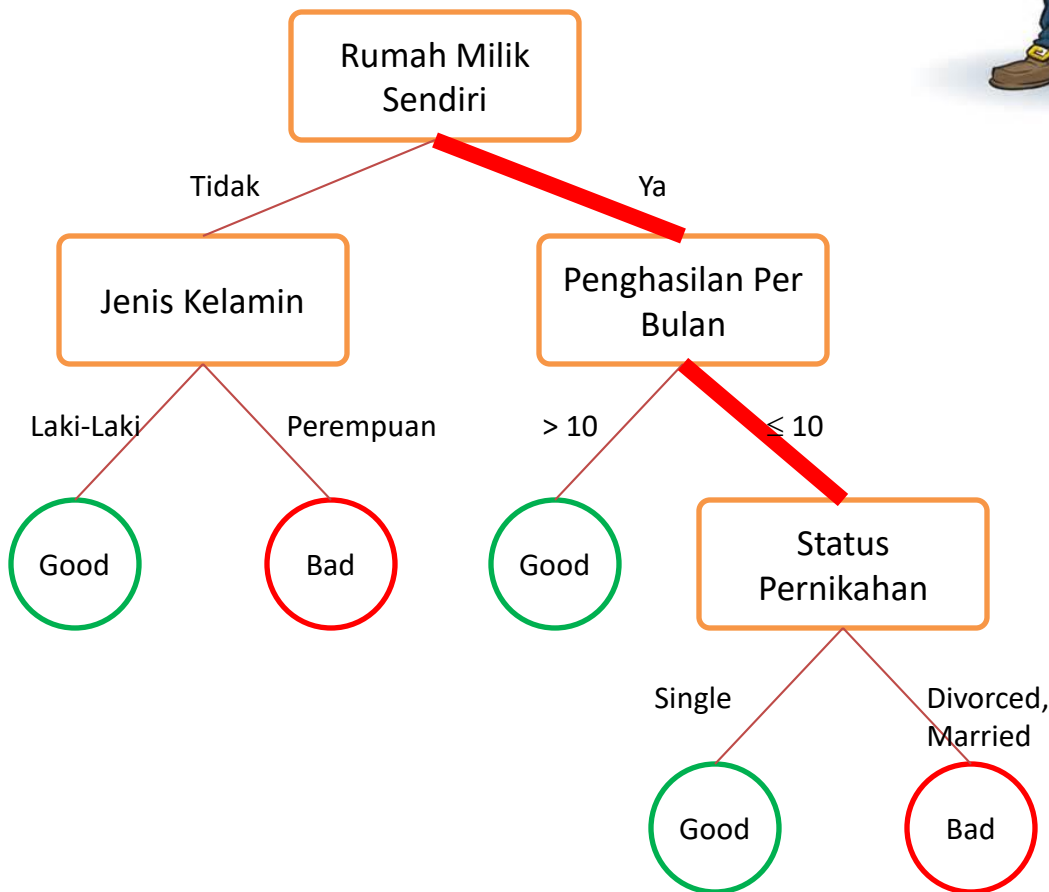


Ilustrasi penggunaan pohon klasifikasi



Profil:

- Pria
- **Rumah Sendiri**
- **Penghasilan 8 juta per bulan**
- Bujangan

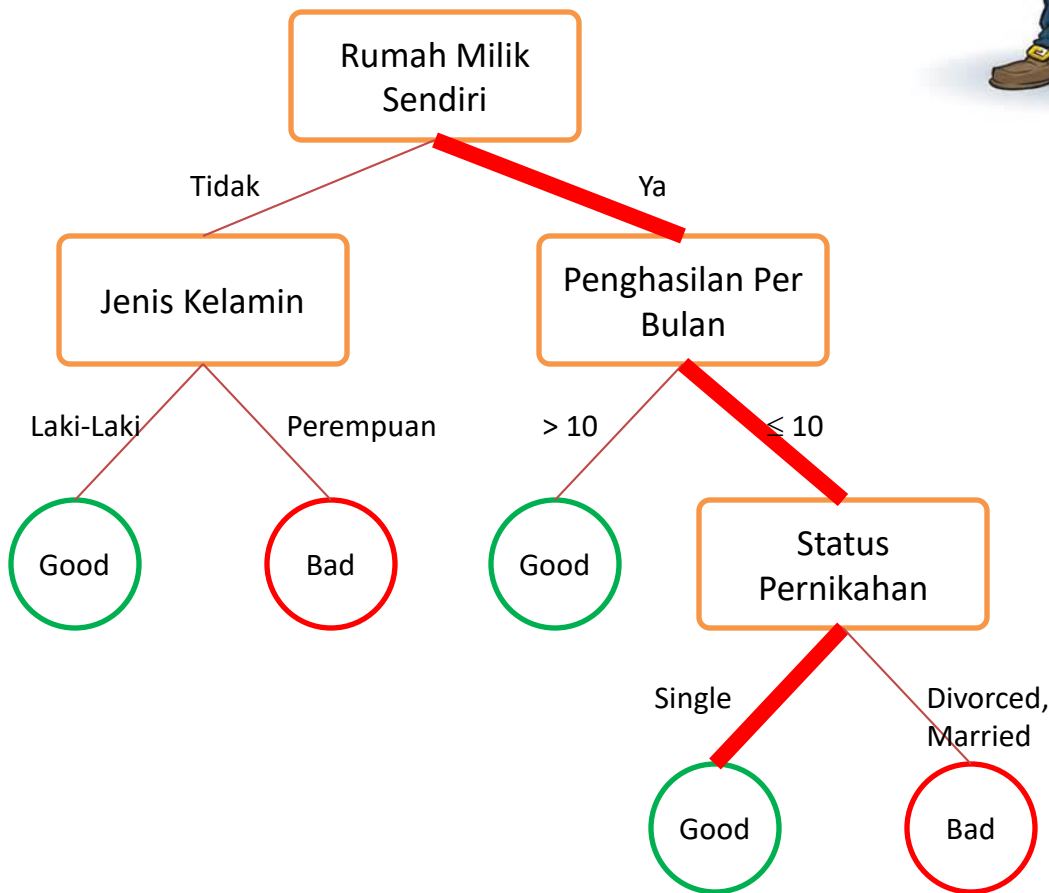


Ilustrasi penggunaan pohon klasifikasi



Profil:

- Pria
- **Rumah Sendiri**
- **Penghasilan 8 juta per bulan**
- **Bujangan**

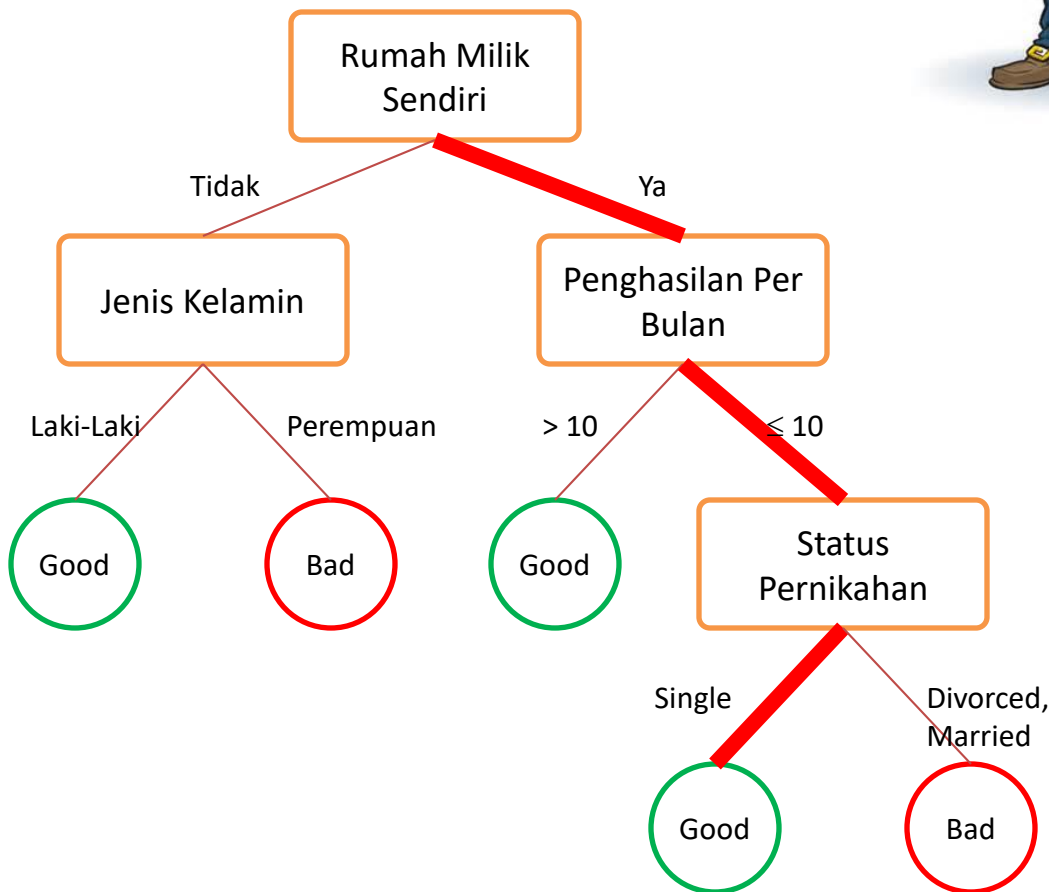


Ilustrasi penggunaan pohon klasifikasi



Profil:

- Pria
- **Rumah Sendiri**
- **Penghasilan 8 juta per bulan**
- **Bujangan**



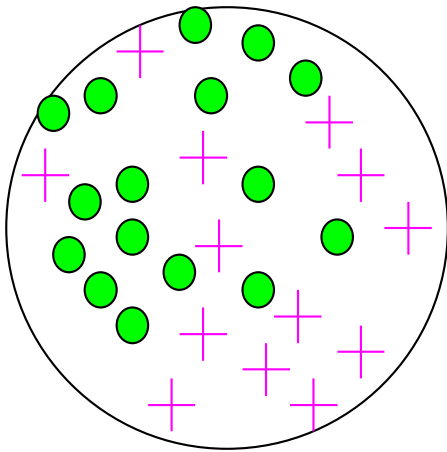
Entropy dan Information Gain

Entropy dan Information Gain

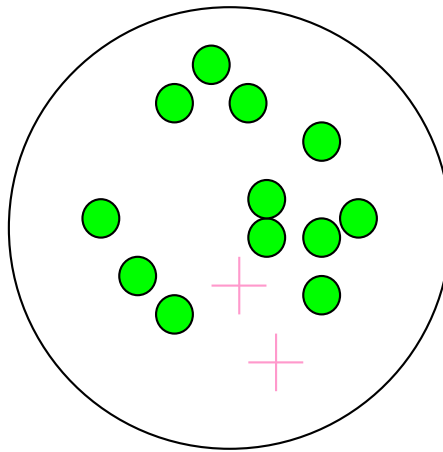
- Andaikan sebuah gugus data D berisi individu-individu dengan dua kelas yaitu kelas YES dan NO, dengan proporsi yang YES sebesar p , dan tentusaja $(1 - p)$ lainnya tergolong kelas NO.
- Entropi dari gugus data tersebut adalah
$$E(D) = -p \log_2(p) - (1-p) \log_2(1-p)$$
- Gugus data yang seluruh amatannya dari kelas YES akan memiliki $E(D) = 0$
- Gugus data yang seluruh amatannya dari kelas NO juga akan memiliki $E(D) = 0$
- Entropi ini adalah ukuran kehomogenan data (impurity)

Entropy dan Information Gain

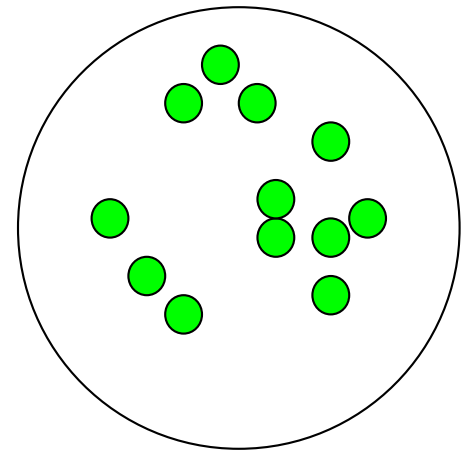
Very impure group



Less impure



Minimum impurity



Entropy dan Information Gain

- Andaikan sebuah gugus data D dibagi menjadi beberapa kelompok, misalnya D_1, D_2, \dots, D_k berdasarkan variabel prediktor V
- Dari setiap D_i bisa dihitung entropinya, yaitu $E(D_i)$

- Information Gain adalah

$$IG(D, V) = E(D) - \sum_{i=1}^k \frac{|D_i|}{|D|} E(D_i)$$

- Variabel prediktor yang kuat hubungannya akan memiliki information gain yang semakin besar

Entropy dan Information Gain

Frequency Percent Row Pct Col Pct	Table of Jenis_Kelamin by Tertarik_Beli			
	Jenis_Kelamin(Jenis Kelamin)	Tertarik_Beli(Tertarik Beli)		
		tidak	tertarik	Total
perempuan		561	27	588
		51.75	2.49	54.24
		95.41	4.59	
		74.80	8.08	
laki-laki		189	307	496
		17.44	28.32	45.76
		38.10	61.90	
		25.20	91.92	
Total		750	334	1084
		69.19	30.81	100.00

$$\begin{aligned}
 E(\text{TOTAL}) &= -p \log_2(p) - (1-p) \log_2(1-p) \\
 &= -0.3081 \log_2(0.3081) - 0.6919 \log_2(0.6919) \\
 &= 0.8910
 \end{aligned}$$

$$\begin{aligned}
 E(\text{Perempuan}) &= -p \log_2(p) - (1-p) \log_2(1-p) \\
 &= -0.0459 \log_2(0.0459) - 0.9541 \log_2(0.9541) \\
 &= 0.2688
 \end{aligned}$$

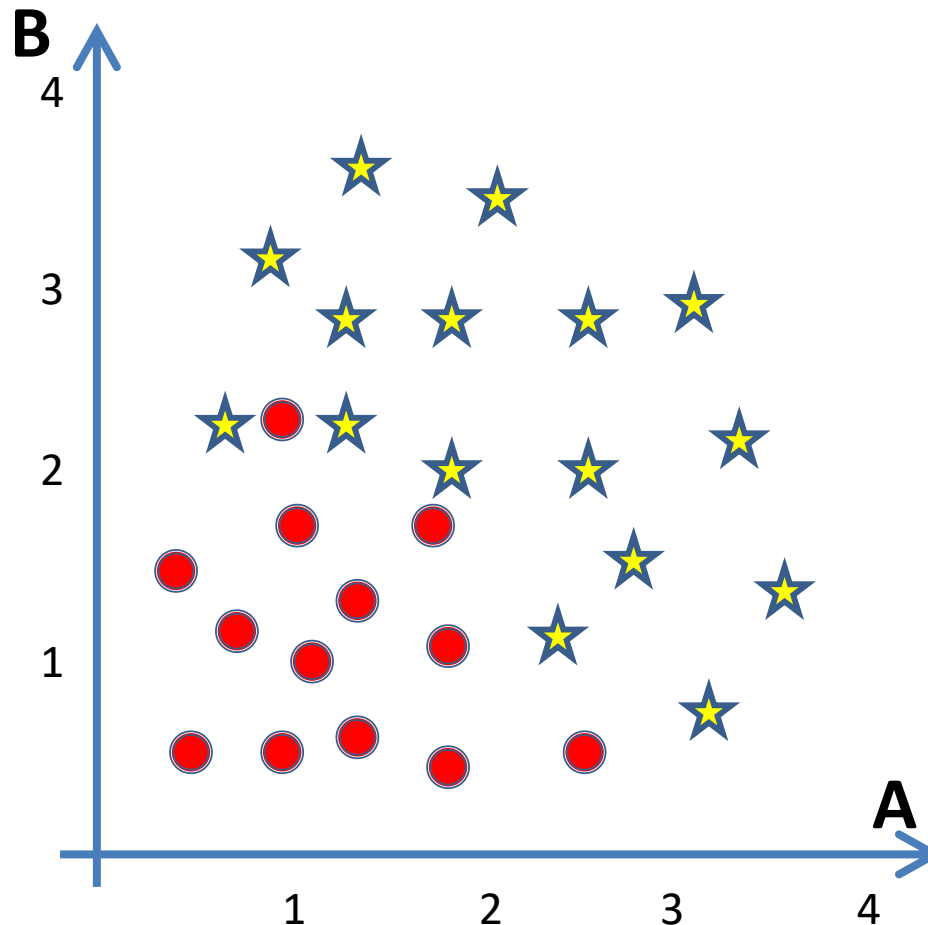
$$\begin{aligned}
 E(\text{Laki-Laki}) &= -p \log_2(p) - (1-p) \log_2(1-p) \\
 &= -0.6190 \log_2(0.6190) - 0.3810 \log_2(0.3810) \\
 &= 0.9588
 \end{aligned}$$

Information Gain dari Variabel Jenis Kelamin

$$\begin{aligned}
 IG &= 0.8910 - (588/1084 * 0.2688 + 496/1084 * 0.9588) \\
 &= 0.8910 - 0.5845 \\
 &= 0.3065
 \end{aligned}$$

Pohon Klasifikasi

Ide Dasar

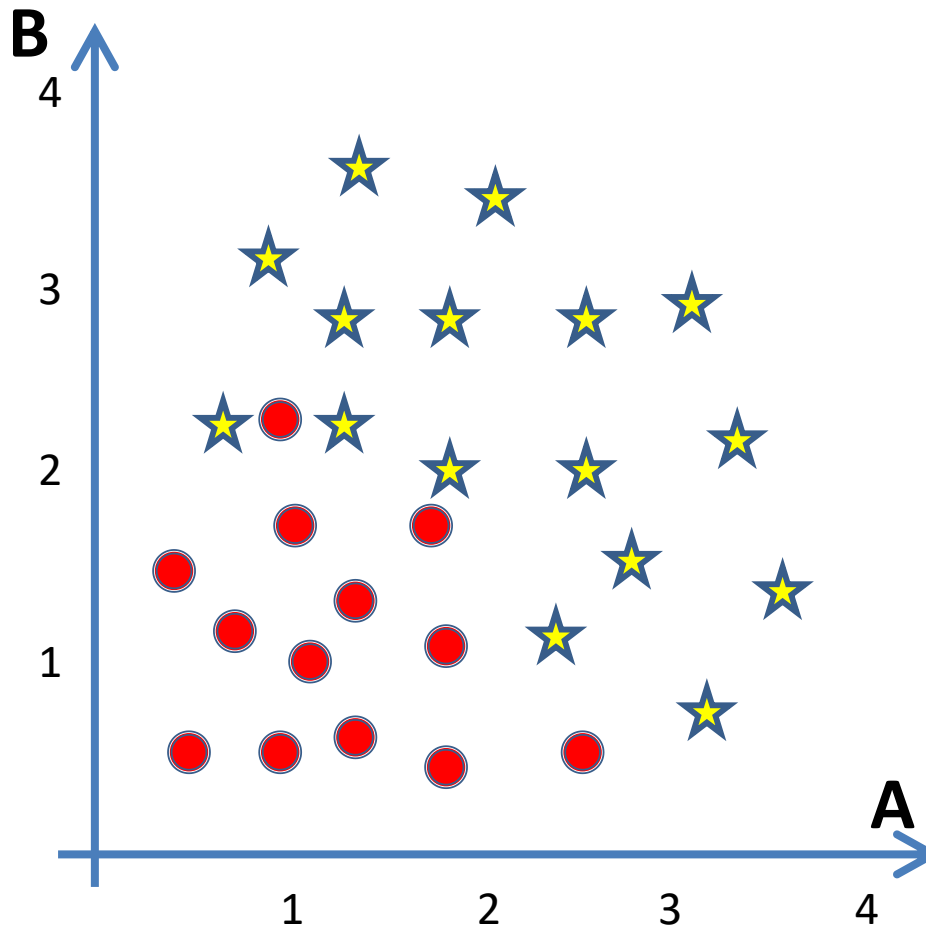


★ 16 obs
● 13 obs

Mencari pemisah
terbaik antara
individu ★
dengan individu ●

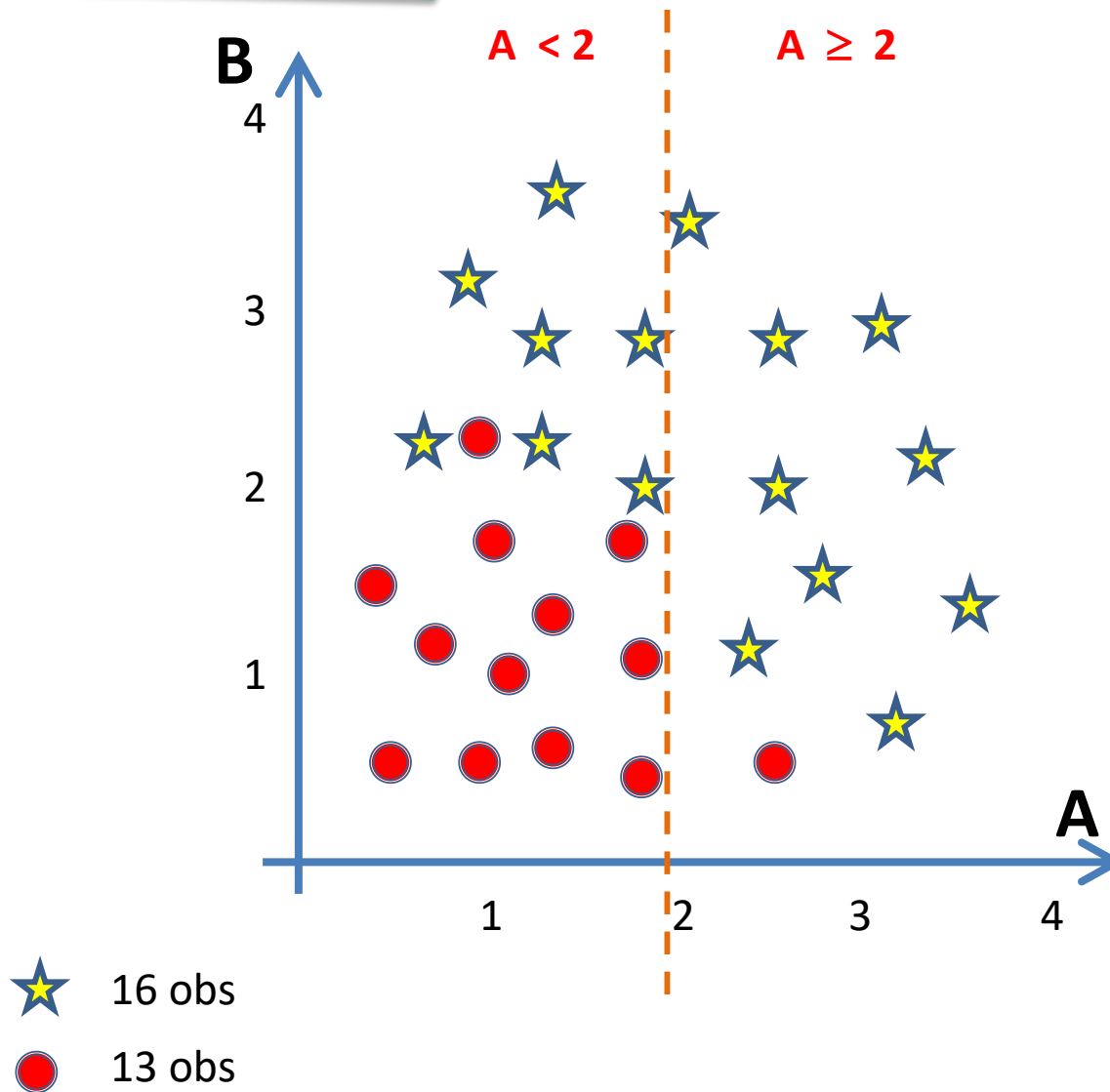
Pemisahan dilakukan
untuk masing-masing
variabel, bukan
kombinasinya.

Ide Dasar



Pemisah yang dicari adalah yang menyebabkan data hasil pemisahannya bersifat homogen kelasnya.

Ide Dasar



Pemisahan menggunakan garis $A = 2$, menghasilkan dua kelompok:

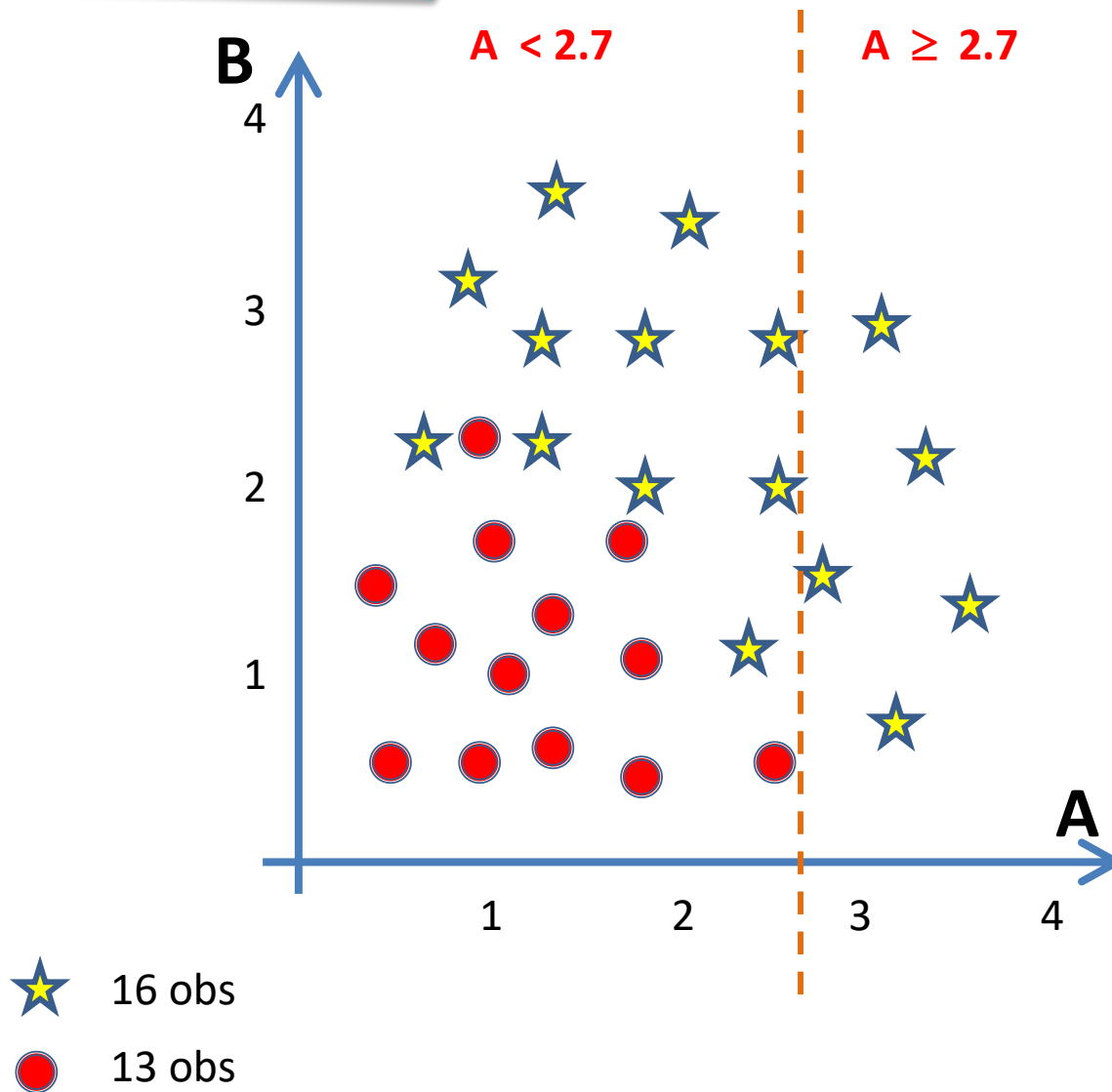
Kelompok 1 $A < 2$

★ 7 obs
● 12 obs

Kelompok 2 $A \geq 2$

★ 9 obs
● 1 obs

Ide Dasar



Pemisahan menggunakan garis $A = 2.7$, menghasilkan dua kelompok:

Kelompok 1 $A < 2.7$

★ 11 obs

● 13 obs

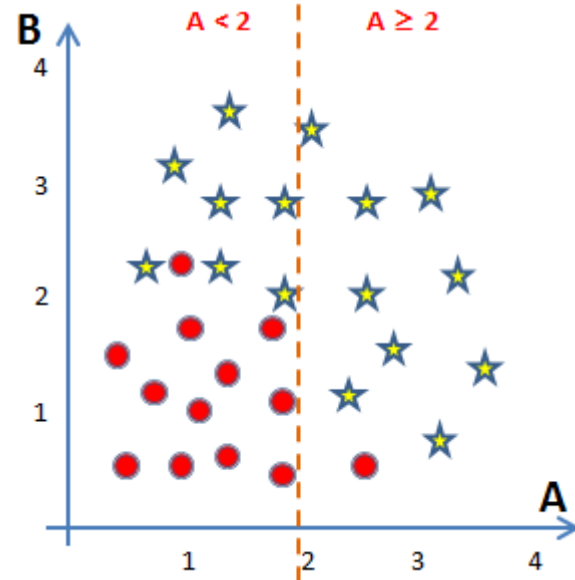
Kelompok 2 $A \geq 2.7$

★ 5 obs

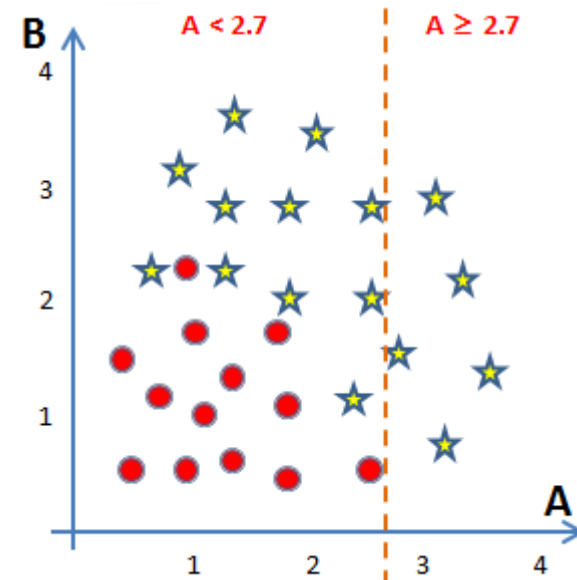
● 0 obs

Ide Dasar

★ 16 obs
● 13 obs



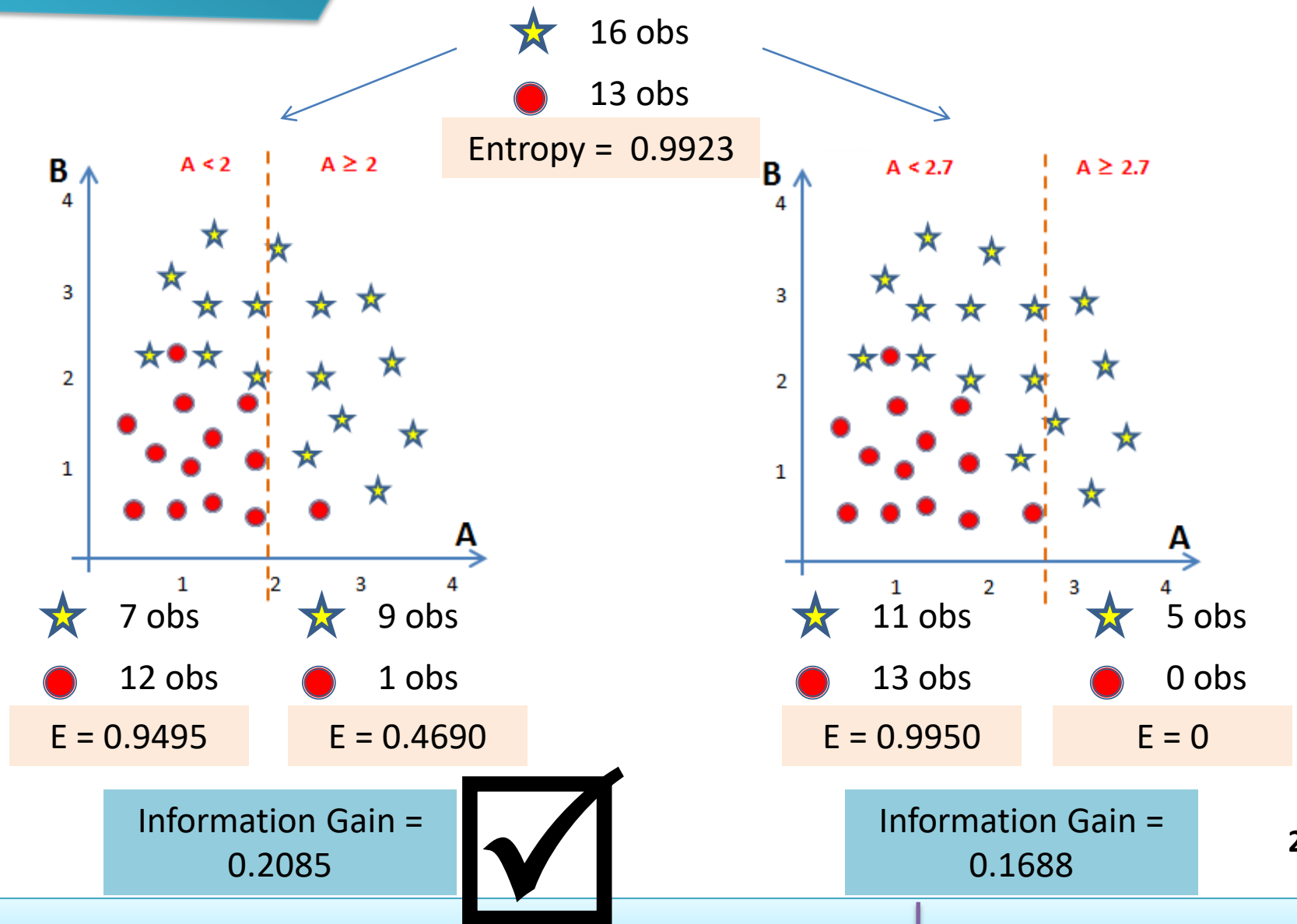
★ 7 obs ★ 9 obs
● 12 obs ● 1 obs



★ 11 obs ★ 5 obs
● 13 obs ● 0 obs

Mana yang lebih baik?

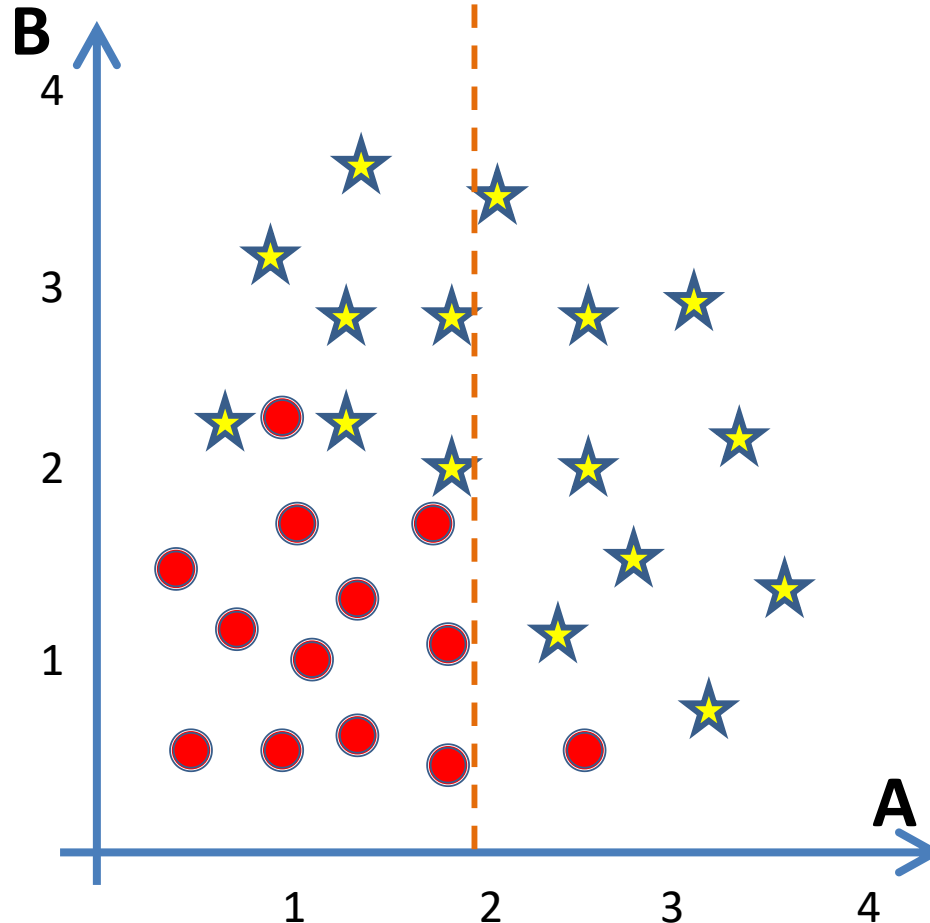
Ide Dasar



Ide Dasar

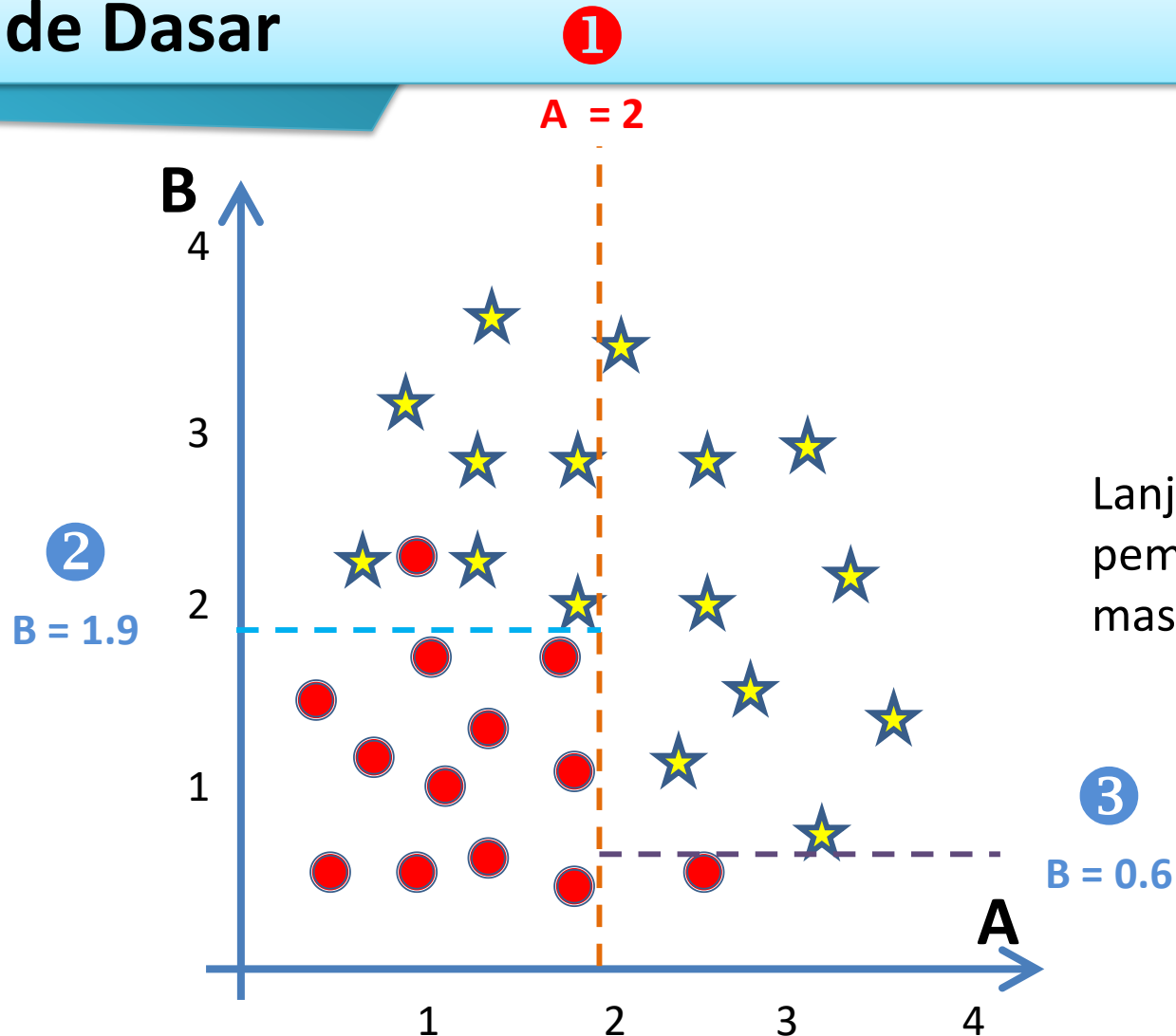
1

$A = 2$



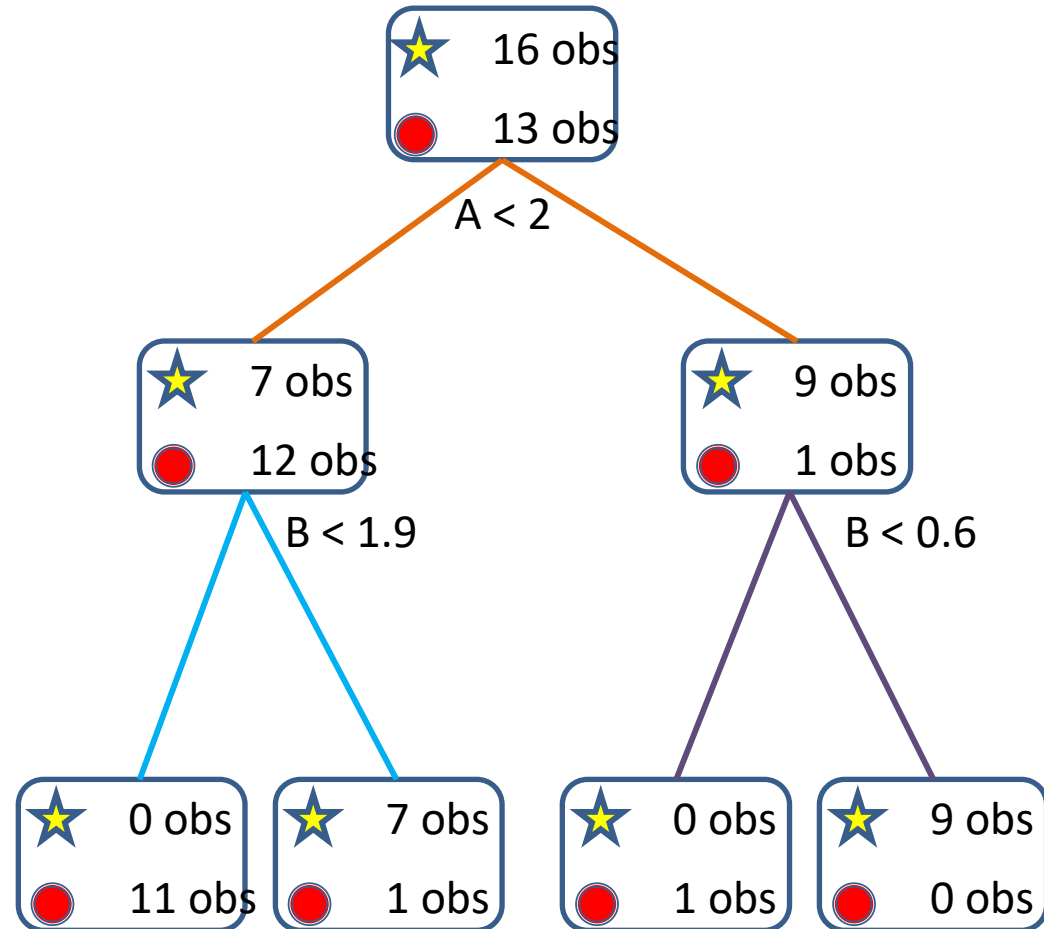
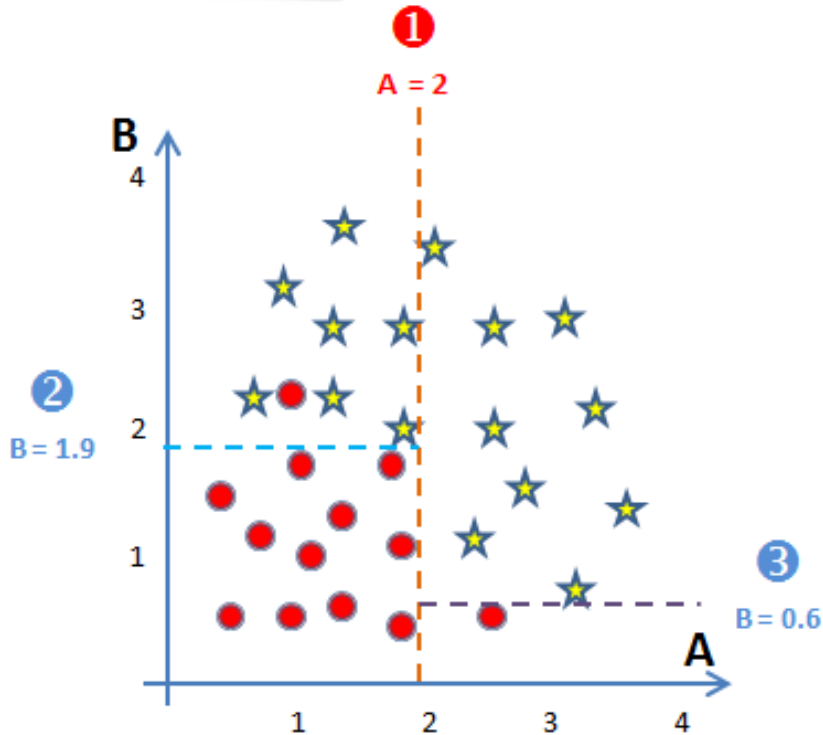
Lanjutkan mencari pemisahan untuk masing-masing kelompok....

Ide Dasar

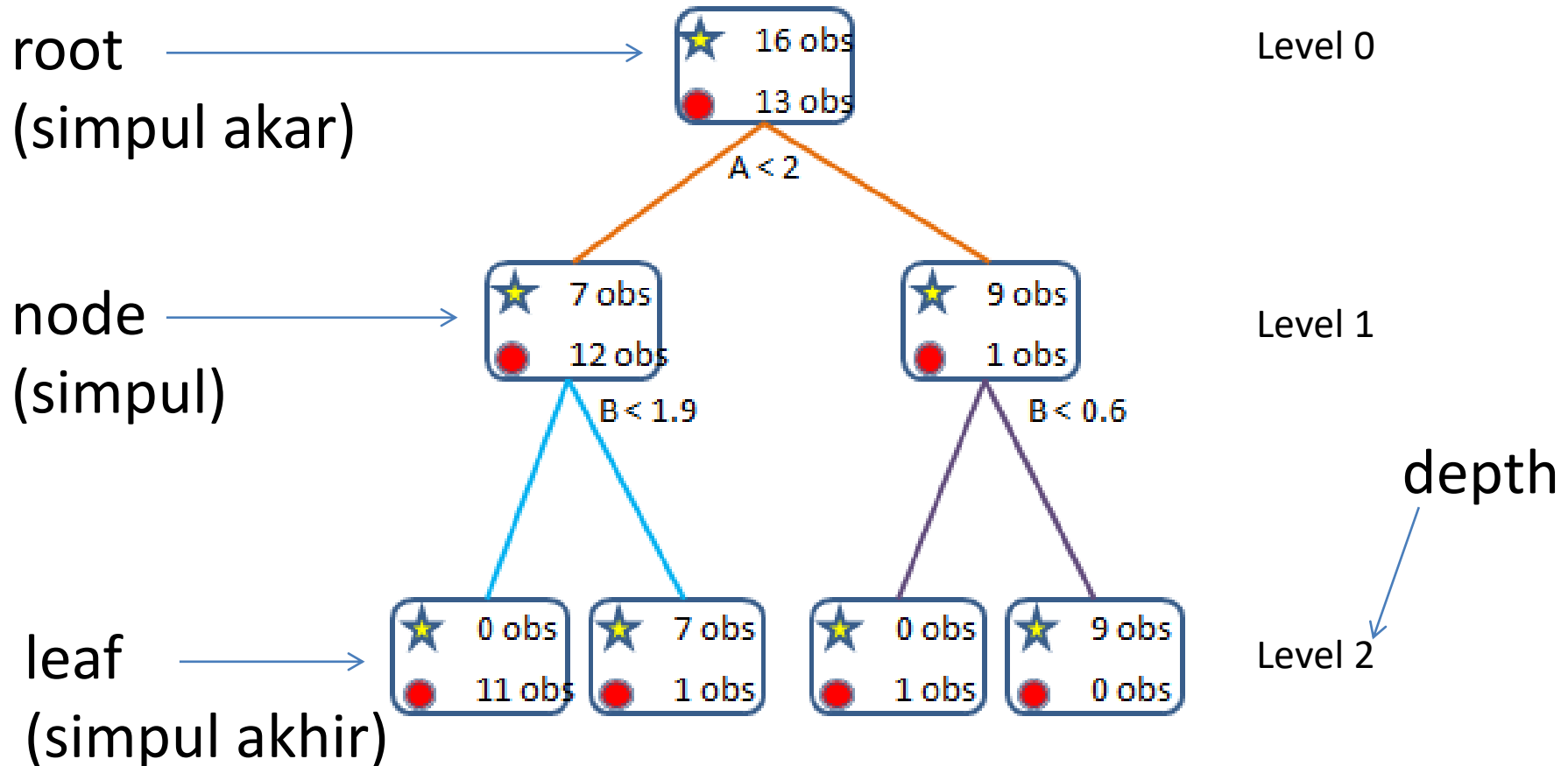


Lanjutkan mencari pemisahan untuk masing-masing kelompok....

Representasi Hasil Pemisahan



Beberapa Istilah



Kriteria Penghentian Proses Pemisahan

- Simpul berisi amatan yang berasal dari satu kelas variabel respon
- Simpul berisi amatan yang seluruh variabel prediktornya identik
- Simpul berisi amatan yang kurang dari ukuran simpul minimal yang ditentukan di awal
- Kedalaman pohon sudah mencapai kedalaman maksimal

Ilustrasi: Data lowbwt.csv

<https://github.com/bagusco/fkh>

- x nomor urut
- id Identification Code
- low Low birth weight (1: ≥ 2500 , 2: < 2500 g)
- age Age of mother (Years)
- lwt Weight of mother at last menstrual period (Pounds)
- race Race (1: White, 2: Black, 3: Other)
- smoke Smoking status during pregnancy (1: No, 2: Yes)
- ptl History of premature labor (1: None, 2: One, 3: Two, etc)
- ht History of hypertension (1: No, 2: Yes)
- ui Presence of Uterine irritability (1: No, 2: Yes)
- ftv Number of physician visits during the first trimester (1: None, 2: One, 3: Two, etc)
- bwt Recorded birth weight (Grams)

Ilustrasi Sederhana

- Gunakan “data tree.csv”
- Variabel :
 - "No"
 - "Jenis.Kelamin"
 - "Single"
 - "Tinggal.di.Kota"
 - "usia"
 - "Perokok"
 - "Budget"
 - "Kesukaan"
 - "Tertarik.Beli."

Ilustrasi Sederhana

```
setwd("D:/bagusco/bagusco/Kuliah S2 --- Pemodelan Klasifikasi/Genap 2017 2018")  
data <- read.csv("data tree.csv")
```

```
data$tertarik <- factor(data$Tertarik.Beli., levels = c(0, 1), labels=c("tidak", "tertarik"))  
data$jk <- factor(data$Jenis.Kelamin, levels=c(0,1), labels=c("p", "l"))  
data$tempattinggal <- factor(data$Tinggal.di.Kota, levels = c(0,1), labels = c("desa", "kota"))  
data$single <- factor(data$Single, levels = c(0,1), labels = c("Menikah", "Single"))  
data$merokok <- factor(data$Perokok, levels = 0:1, labels = c("Tidak", "Ya"))
```

```
setwd("D:/bagusco/bagusco/Kuliah S2 --- Pemodelan Klasifikasi/Genap 2017 2018")  
data <- read.csv("data tree.csv")
```

```
library(discretization)  
entropy_total <- ent(data$tertarik)
```

```
entropy_lakilaki <- ent(data$tertarik[data$jk == "l"])  
entropy_perempuan <- ent(data$tertarik[data$jk == "p"])  
IG_jk <- entropy_total - length(data$tertarik[data$jk == "l"])*entropy_lakilaki / nrow(data) -  
length(data$tertarik[data$jk == "p"])*entropy_perempuan / nrow(data)
```

IG_jk → 0.21

```
entropy_merokok <- ent(data$tertarik[data$merokok == "Ya"])  
entropy_tidakmerokok <- ent(data$tertarik[data$merokok == "Tidak"])  
IG_merokok <- entropy_total - length(data$tertarik[data$merokok ==  
"Ya"])*entropy_merokok / nrow(data) - length(data$tertarik[data$merokok ==  
"Tidak"])*entropy_tidakmerokok / nrow(data)
```

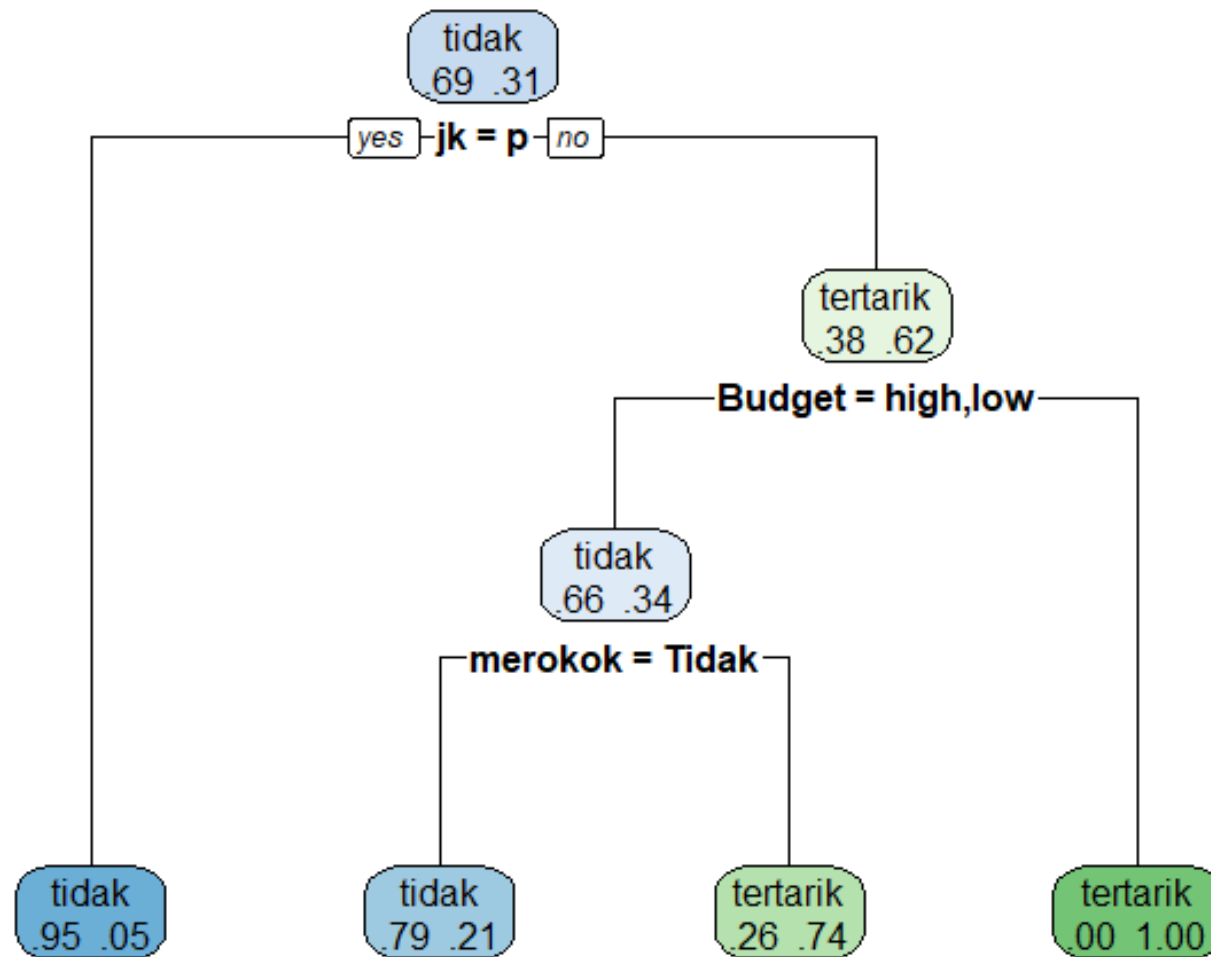
IG_merokok → 0.07

Ilustrasi Sederhana

```
library(rpart)
library(rpart.plot)

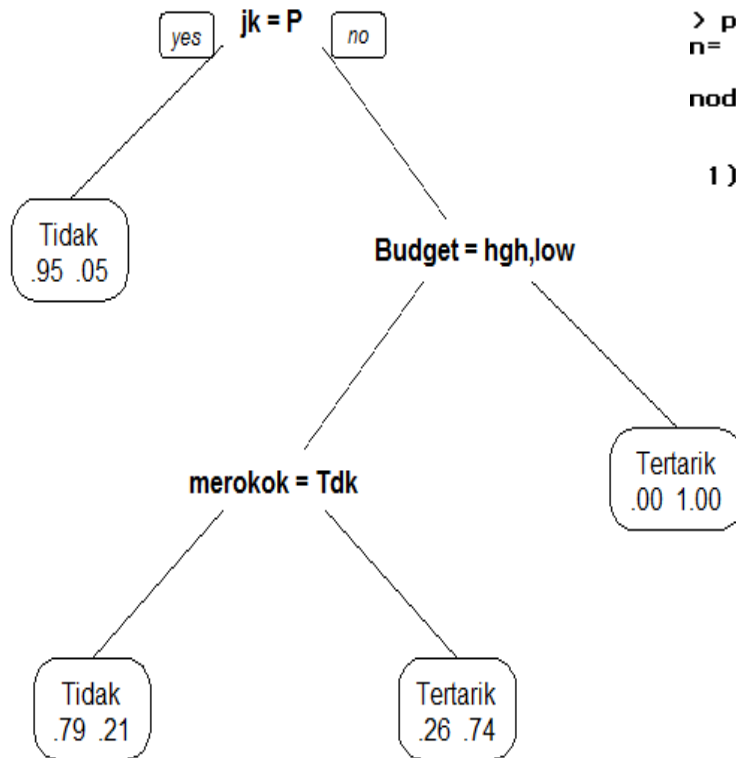
model = rpart(tertarik ~ jk + tempattinggal + single + usia + merokok + Budget,
              data = data, method="class",
              control = rpart.control(minsplit = 100, cp = 0))
print(model)
rpart.plot(model, extra=4)
```

Grafik



Ilustrasi Sederhana

```
model = rpart(tertarik ~ jk + kota + single + usia + merokok + Budget,  
data = a.data, method="class",  
control = rpart.control(minsplit = 100, cp = 0))
```



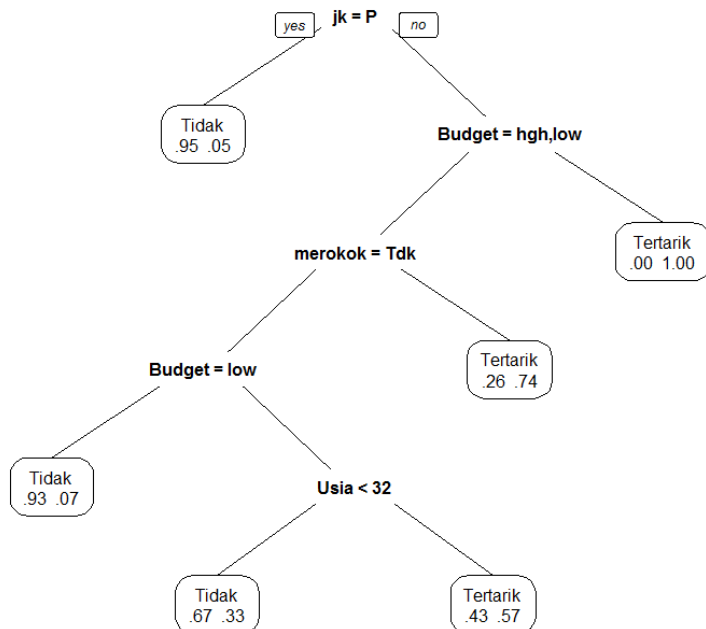
```
> print(model)
n= 1084

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 1084 334 Tidak (0.691881919 0.308118081)
2) jk=P 588 27 Tidak (0.954081633 0.045918367) *
3) jk=L 496 189 Tertarik (0.381048387 0.618951613)
6) Budget=high,low 283 95 Tidak (0.664310954 0.335689046)
12) merokok=Tidak 217 46 Tidak (0.788018433 0.211981567) *
13) merokok=Ya 66 17 Tertarik (0.257575758 0.742424242) *
7) Budget=medium 213 1 Tertarik (0.004694836 0.995305164) *
```

Ilustrasi Sederhana

```
model = rpart(tertarik ~ jk + kota + single + usia + merokok + Budget,  
data = a.data, method="class",  
control = rpart.control(minsplit = 50, cp = 0))
```



```
> print(model)
n= 1084

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 1084 334 Tidak (0.691881919 0.308118081)
2) jk=P 588 27 Tidak (0.954081633 0.045918367) *
3) jk=L 496 189 Tertarik (0.381048387 0.618951613)
6) Budget=high,low 283 95 Tidak (0.666310954 0.335689046)
12) merokok=Tidak 217 46 Tidak (0.788018433 0.211981567)
24) Budget=low 147 11 Tidak (0.925170068 0.074829932) *
25) Budget=high 70 35 Tidak (0.500000000 0.500000000)
50) Usia< 31.5 21 7 Tidak (0.666666667 0.333333333) *
51) Usia>=31.5 49 21 Tertarik (0.428571429 0.571428571) *
13) merokok=Ya 66 17 Tertarik (0.257575758 0.742424242) *
7) Budget=medium 213 1 Tertarik (0.004694836 0.995305164) *
```

Menilai Keباikan Pohon Klasifikasi

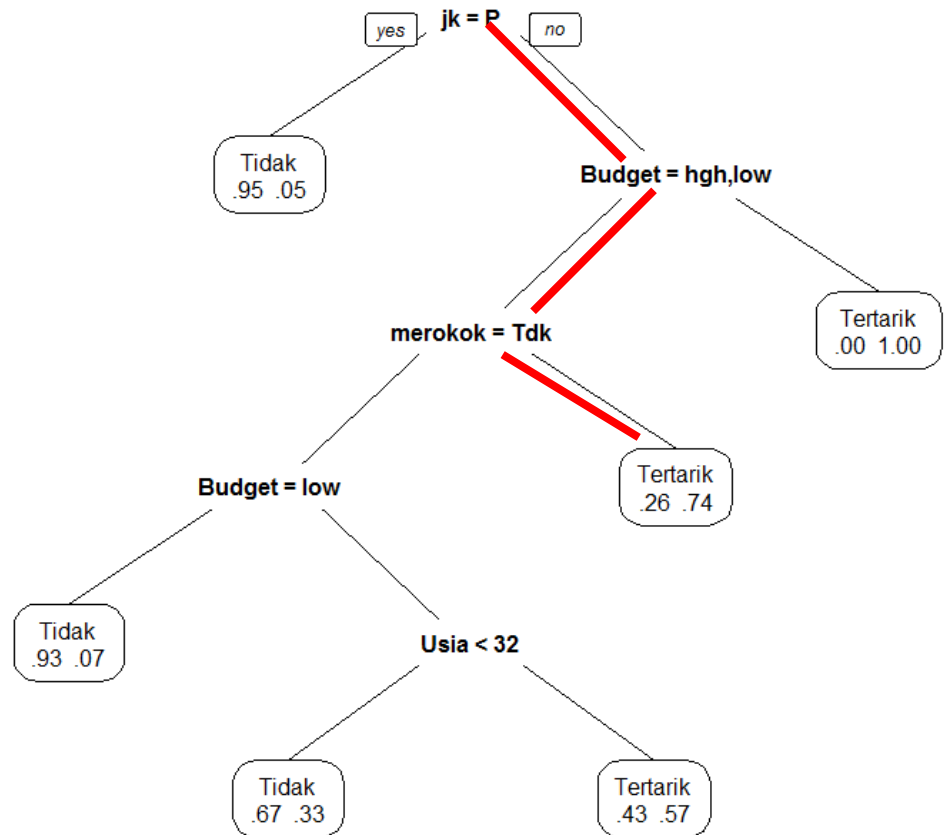
Prediksi Variabel Respon

- Untuk setiap individu yang diketahui nilai-nilai variabel prediktor yang muncul pada pohon klasifikasi, kita dapat melakukan prediksi kelas variabel respon. Misalnya jika diketahui usia, jenis kelamin, apakah merokok, dan klasifikasi budget dari seseorang, maka kita dapat memprediksi apakah orang tersebut akan tertarik atau tidak.
- Bagaimana caranya? Gunakan alur pencabangan yang ada pada pohon klasifikasi sampai berhenti di simpul akhir. Berdasarkan simpul akhir itulah kita prediksi dia masuk ke kategori apa.

Prediksi Variabel Respon

Misal

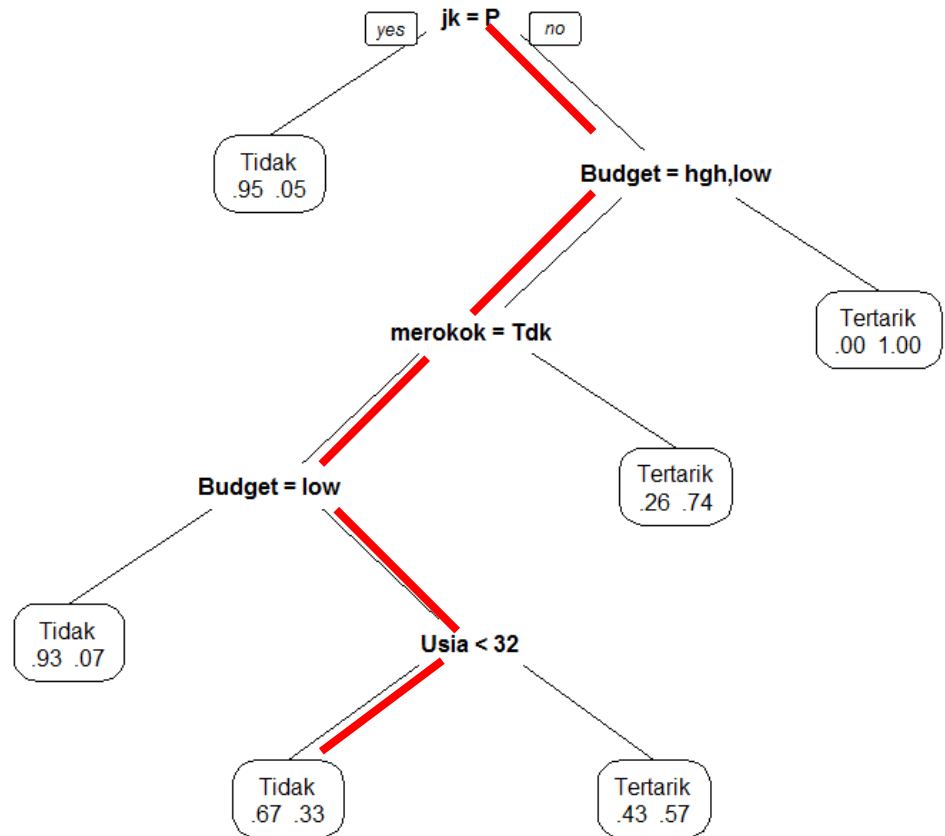
- Jenis Kelamin = Laki-Laki
- Budget Low
- Merokok
- Usia 25 tahun
- Probability TERTARIK = 0.74



Prediksi Variabel Respon

Misal

- Jenis Kelamin = Laki-Laki
- Budget high
- Tidak Merokok
- Usia 25 tahun
- Probability TERTARIK = 0.33



Prediksi Variabel Respon

```
model = rpart(tertarik ~ jk + kota + single + usia + merokok + Budget,  
              data = a.data, method="class",  
              control = rpart.control(minsplit = 50, cp = 0))
```

```
prob_prediksi <- predict(model, newdata=data, type = 'prob')  
head(prob_prediksi, n=10)
```

	Tidak	Tertarik
1	0.925170068	0.07482993
2	0.954081633	0.04591837
3	0.954081633	0.04591837
4	0.954081633	0.04591837
5	0.004694836	0.99530516
6	0.954081633	0.04591837
7	0.004694836	0.99530516
8	0.925170068	0.07482993
9	0.954081633	0.04591837
10	0.428571429	0.57142857

Prediksi Variabel Respon

Andaikan digunakan batasan 0.5 untuk mengelompokkan ketertarikan, sehingga kalau

$\text{Prob}(\text{Tertarik}) > 0.5 \rightarrow \text{Tertarik}$

$\text{Prob}(\text{Tertarik}) \leq 0.5 \rightarrow \text{tidak}$

Maka kita akan dapatkan

	Tidak	Tertarik		Prediksi
1	0.925170068	0.07482993	→	Tidak
2	0.954081633	0.04591837	→	Tidak
3	0.954081633	0.04591837	→	Tidak
4	0.954081633	0.04591837	→	Tidak
5	0.004694836	0.99530516	→	Tertarik
6	0.954081633	0.04591837	→	Tidak
7	0.004694836	0.99530516	→	Tertarik
8	0.925170068	0.07482993	→	Tidak
9	0.954081633	0.04591837	→	Tidak
10	0.428571429	0.57142857	→	Tertarik

Prediksi Variabel Respon

Perbandingan antara respon yang sebenarnya dengan dugaan

	Tertarik_beli	dugaan
1	Tidak	Tidak
2	Tidak	Tidak
3	Tidak	Tidak
4	Tidak	Tidak
5	Tertarik	Tertarik
6	Tidak	Tidak
7	Tertarik	Tertarik
8	Tidak	Tidak
9	Tidak	Tidak
10	Tidak	Tertarik → salah prediksi

Kebaikan pohon klasifikasi

Kebaikan dapat dilihat dari seberapa tinggi kemampuan pohon klasifikasi menghasilkan dugaan yang sama dengan kondisi yang sesungguhnya.

```
prediksi <- ifelse (prob_prediksi[,2] > 0.5, "tertarik", "tidak")  
table(data$tertarik, prediksi)
```

	prediksi	
	tertarik	tidak
tidak	39	711
tertarik	289	45

Kebaikan pohon klasifikasi

```
library(caret)
confusionMatrix(prediksi, data$tertarik,
positive="tertarik")
```

Confusion Matrix and Statistics

	Reference	
Prediction	tidak	tertarik
tidak	711	45
tertarik	39	289

Accuracy : 0.9225

95% CI : (0.905, 0.9377)

No Information Rate : 0.6919

P-Value [Acc > NIR] : <2e-16

Kappa : 0.8173

Mcnemar's Test P-Value : 0.5854

Sensitivity : 0.8653

Specificity : 0.9480

Pos Pred Value : 0.8811

Neg Pred Value : 0.9405

Prevalence : 0.3081

Detection Rate : 0.2666

Detection Prevalence : 0.3026

Balanced Accuracy : 0.9066

'Positive' Class : tertarik

Kebaikan pohon klasifikasi

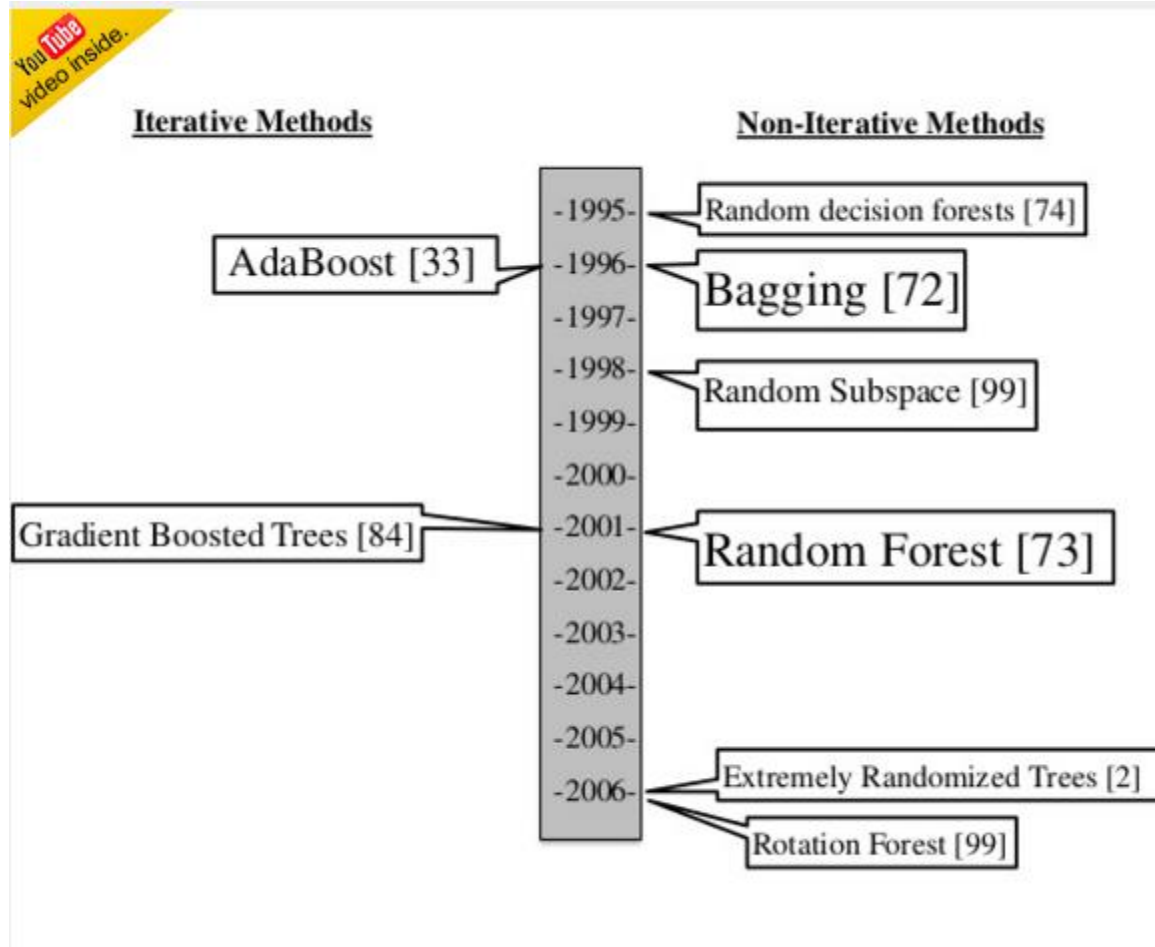
Cut-Off	Accuracy	Sensitivity	Specificity
0.3	91.60%	88.62%	92.93%
0.5	92.25%	86.53%	94.80%
0.6	91.60%	78.14%	97.60%

Perkembangan Lebih Lanjut

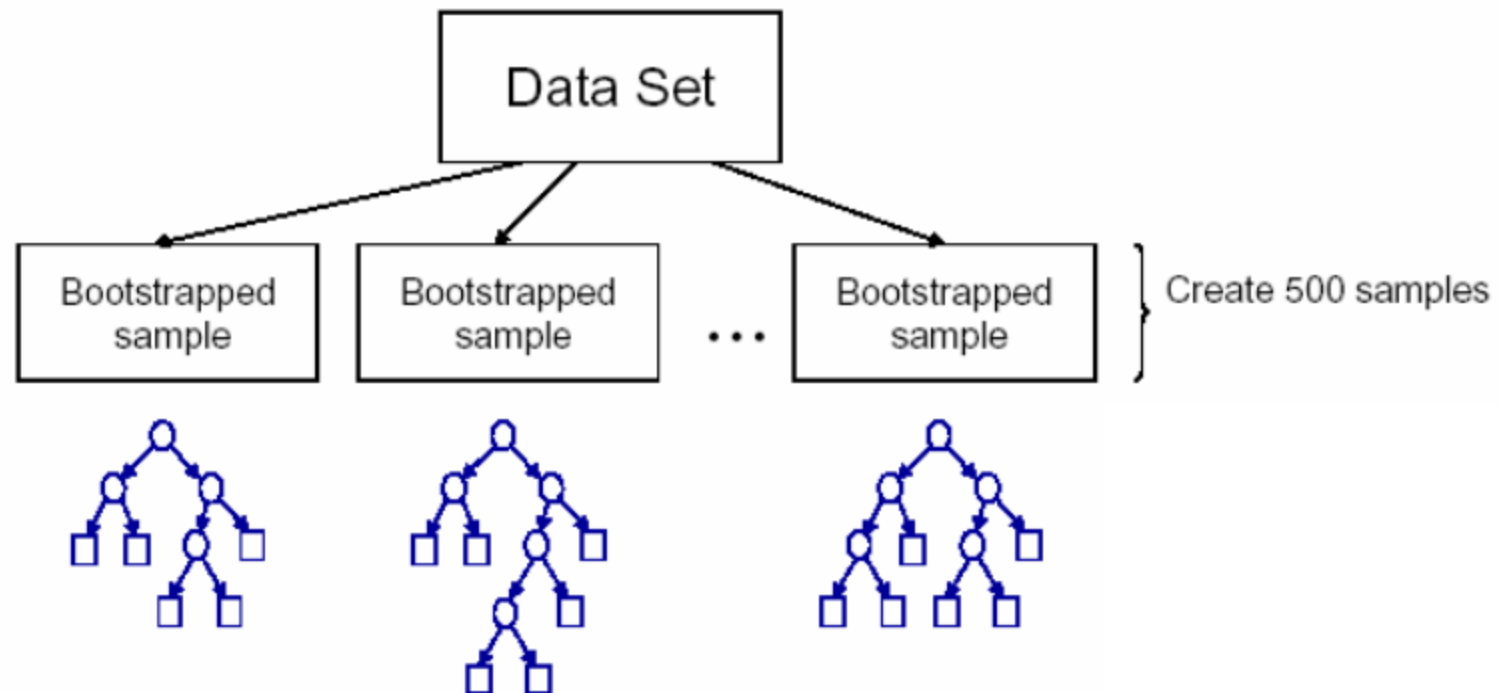
Perkembangan Lebih Lanjut: Ensemble Tree

- Bagging (Bootstrap Aggregating)
 - Breiman, L (1996). Bagging predictors. *Machine Learning* 24 (2): 123–140
- Boosting
 - Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *ICML* (Vol. 96, pp. 148-156).
- Random Forest
 - Breiman L (2001). Random Forests. *Machine Learning*, 45, 5-32.
 - Ho, Tin Kam (1995). [Random Decision Forests](#). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282
 - Ho, Tin Kam (1998). ["The Random Subspace Method for Constructing Decision Forests"](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (8): 832–844
- Rotation Forest
 - Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10), 1619-1630.

Perkembangan Lebih Lanjut

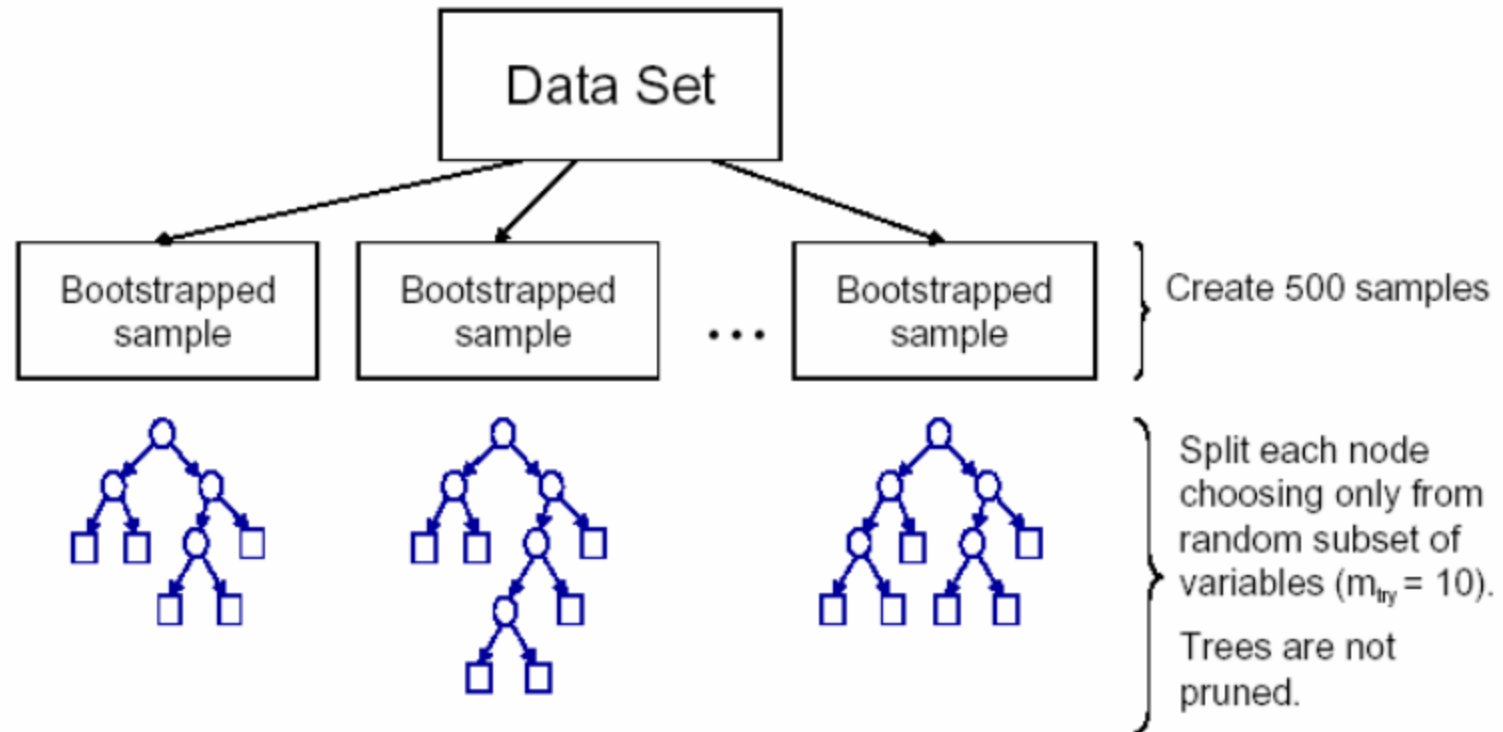


Bagging



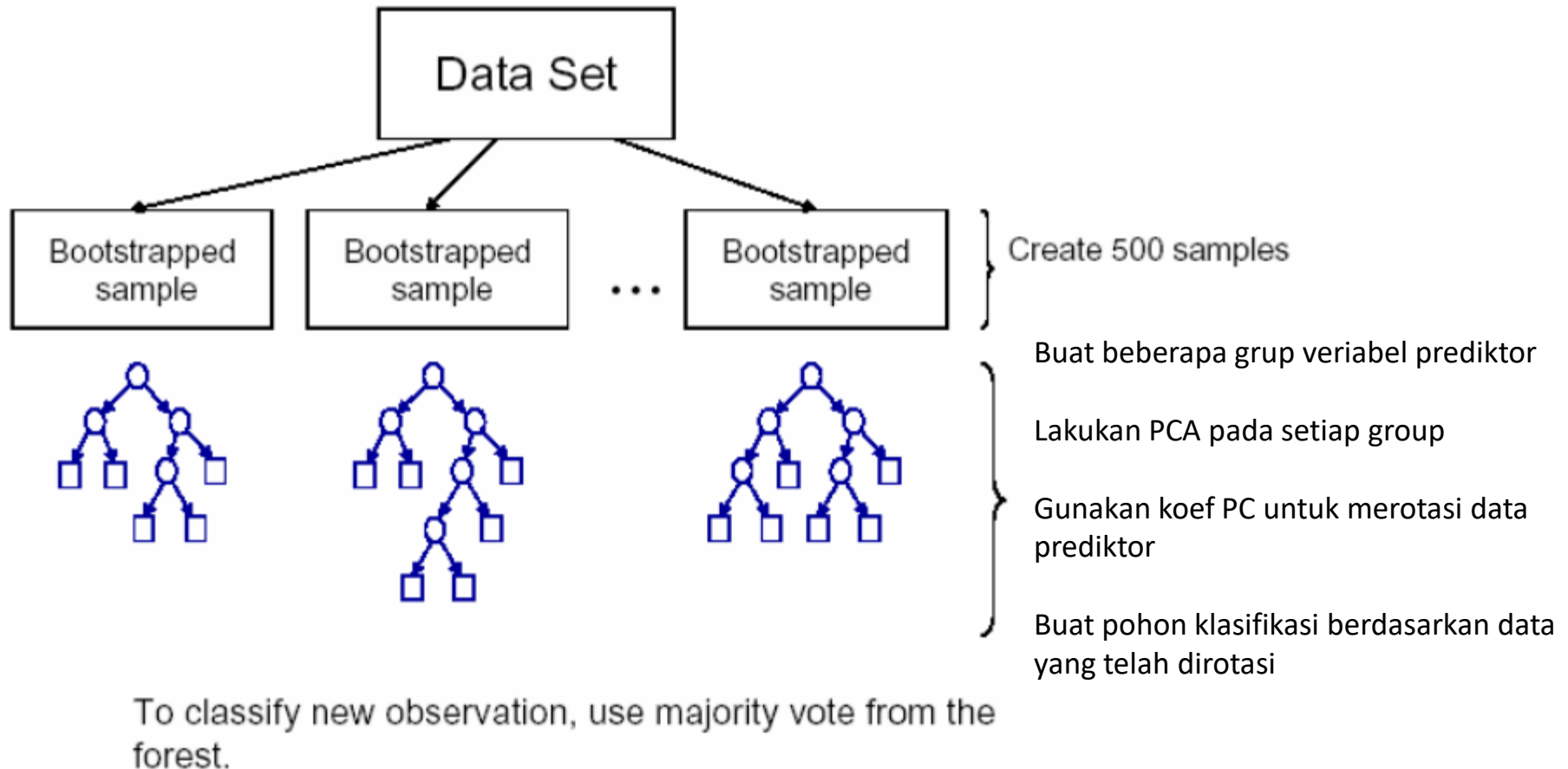
To classify new observation, use majority vote from the forest.

Random Forest



To classify new observation, use majority vote from the forest.

Rotation Forest



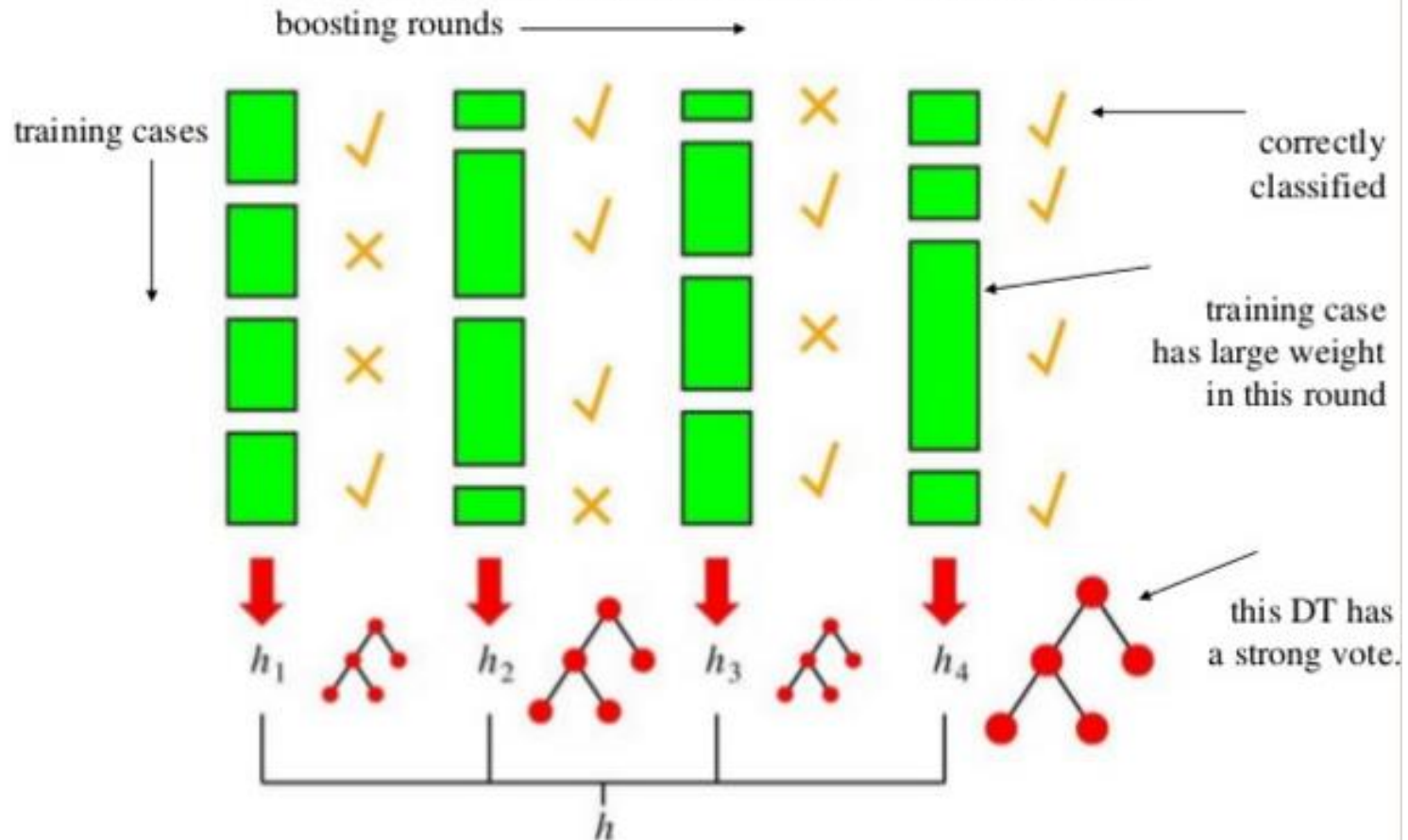
Boosting

YouTube
video inside.

AdaBoost

(Freund & Schapire, 1996)

"Best off-the-shelf classifier in the world" – Breiman (1996)



Empirical study

- Pendekatan ensemble tree menghasilkan prediksi yang lebih baik dibandingkan pohon klasifikasi tunggal
- Ensemble tree banyak digunakan untuk menangani masalah-masalah:
 - Ketidakseimbangan Kelas (Imbalanced Class)
 - Curse of Dimensionality
 - Klasifikasi Multi-Kelas

Beyond Classification Tasks

- Regression tree (Breiman *et al.*, 1984)
- Survival tree (Bou-Hamad *et al.*, 2011)
- Clustering tree (Blockeel *et al.*, 1998)
- Recommendation tree (Gershman *et al.*, 2010):
- Markov model tree (Antwarg *et al.*, 2012)
-



Thank you!

- A data.frame with 618 rows and 18 variables:
- id Subject identifier (1-1503)
- studyyear Year subject entered the study (Year)
- myopic Myopia within the first five years of follow up (1: No, 2: Yes)
- age Age at first visit (Years)
- gender Gender (1: Male, 2: Female)
- spheq Spherical Equivalent Refraction (diopter)
- al Axial Length (mm)
- acd Anterior Chamber Depth (mm)
- lt Lens Thickness (mm)
- vcd Vitreous Chamber Depth (mm)
- sporthr How many hours per week outside of school the child spent engaging in sports/outdoor activities (Hours per week)
- readhr How many hours per week outside of school the child spent reading for pleasure (Hours per week)
- compthr How many hours per week outside of school the child spent playing video/computer games or working on the computer (Hours per week)
- studyhr How many hours per week outside of school the child spent reading or studying for school assignments (Hours per week)
- tvhr How many hours per week outside of school the child spent watching television (Hours per week)
- diopterhr Composite of near-work activities (Hours per week)
- mommy Was the subject's mother myopic? (1: No, 2: Yes)
- dadmy Was the subject's father myopic? (1: No, 2: Yes)

Ensemble Learning

disusun oleh:

Bagus Sartono

bagusco@gmail.com

0852-1523-1823

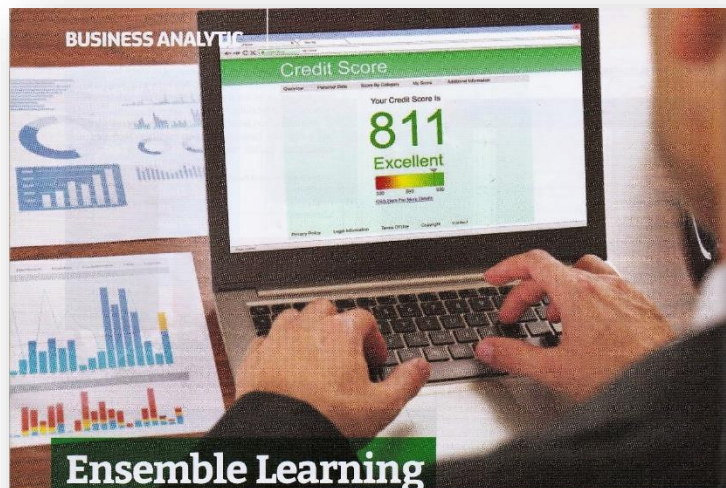


Departemen Statistika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Pertanian Bogor

2019



Ensemble Learning Primadona Analitik di Masa Depan

Salah satu aktivitas penting dalam proses analitik adalah menghasilkan model prediktif, yaitu suatu model yang diharapkan dapat memberikan prediksi yang sangat baik terhadap kejadian di masa mendatang.

Model Prediktif

Model prediktif tersebut antara lain diperlukan oleh bank dan perusahaan pembiayaan dalam bentuk *credit scoring model*. Tujuannya, memperkirakan apakah seseorang yang mengajukan aplikasi pinjaman akan macet kreditnya atau tidak. Tentu saja, prediksi kreditnya tersebut perlu dilakukan jauh hari sebelum diberikan keputusan apakah aplikasinya ditolak atau diterima. Mereka yang diprediksi akan memiliki peluang besar untuk gagal bayar akan memperoleh skor kecil berdasarkan model yang dibangun. Sebaliknya, yang diprediksi akan mampu membayar dengan lancar diberi skor besar oleh model.

Model-model yang serupa juga diperlukan oleh banyak perusahaan berbasis telemarketing untuk memerlukan *short-list* calon pelanggan untuk dihubungi dan ditawarkan produk. *Short-list* tersebut umumnya diperoleh dari *list* yang sangat panjang dan memuat



Nuzulita Salsita
Mahasiswa Departemen
Statistika FMIPA-IPB
nuzulita@ipb.ac.id
Instagram: nuzulita_01

banyak nama individu. Perusahaan memerlukan model prediktif untuk memisahkan individu yang potensial dan yang tidak. Individu yang potensial adalah mereka yang diprediksi akan menerima tawaran produk yang diajukan oleh petugas telemarketer. Aktivitas ini sangat identik dengan yang dikerjakan dalam kampanye via SMS (*short message service*) oleh berbagai perusahaan retail.

Terdapat banyak pemodelan prediktif untuk melakukan prediksi terjadinya (atau tidak terjadinya) suatu kejadian masa mendatang. Beberapa yang disebut berikut adalah teknik dan algoritma pemodelan yang sering digunakan oleh analis baik yang berbasis pemikiran statistika maupun *machine learning*, yaitu: regresi logistik, analisis diskriminan, *k-nearest-neighbor*, *Bayesian classifier*, *classification tree*, *neural network*, dan *support vector machine*. Ada beberapa algoritma lain yang dapat ditemukan dengan mudah di banyak literatur ilmiah maupun praktis.

Berbagai macam algoritma yang disebutkan di atas dapat digunakan untuk menjawab tujuan sama, dan banyak orang berpendapat bahwa satu sama lain dapat dipandang memiliki sifat *complementary*. Karena itu, kemudian muncul pertanyaan besar: algoritma atau teknik mana yang sebaiknya digunakan? Tidak hanya itu, dengan menerapkan salah satu teknik yang sama, dua orang analis dapat menghasilkan model yang berbeda karena dalam proses pemodelannya dapat saja mereka menggunakan prediktor yang berbeda, menggunakan sampel data yang berbeda, serta menerapkan *pre-processing* yang berbeda sesuai dengan kreativitas masing-masing. Dengan demikian, sekali lagi kemudian muncul pertanyaan: model mana yang sebaiknya digunakan?

Model Selection

Pertanyaan tersebut kemudian berujung pada penggunaan berbagai kriteria untuk menentukan model terbaik. Diskusi kemudian berkembang dalam ranah *model selection* (pemilihan model) yang menggunakan berbagai macam kriteria.

Secara umum, penulis memahami bahwa ada dua kriteria besar dalam pemilihan model mana yang digunakan. Kriteria pertama terkait dengan kinerja prediksinya. Dalam bahasa lain, orang menggunakan istilah akurasi atau ketepatan prediksi. Model dengan akurasi yang lebih tinggi disebut sebagai model yang sebaiknya digunakan. Kriteria ini dikenal sebagai *goodness of fit*. Ukuran yang termasuk dalam kategori ini antara lain *likelihood function*, *correct classification rate*, *sensitivity*, dan *specificity*.

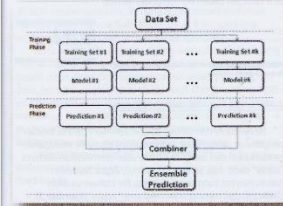
Kriteria yang kedua adalah terkait dengan kesederhanaan model. Secara naluriah, model yang diskaat adalah model yang lebih ringkas, menggunakan *predictor* yang lebih sedikit, atau bentuk-bentuk fungsi yang lebih sederhana. Kriteria kedua ini dikenal sebagai *complexity cost*. Ukuran yang tergolong dalam kriteria ini meliputi banyaknya parameter model, banyaknya simpul pada *tree* dan *neural network*, serta derajat *polynomial* dari variabel *predictor*. *Complexity cost* ini penting diperhatikan agar model prediksi tidak mengalami masalah *overfit*.

Kriteria-kriteria di atas selanjutnya digunakan oleh para analis untuk menentukan model mana yang digunakan. Dua jenis kriteria tersebut banyak digabungkan menjadi satu kriteria gabungan seperti yang dilakukan pada AIC (*Akaike's Information Criterion*) dan yang sejenisnya. Model dengan *goodness-of-fit* besar dan *complexity cost* kecil merupakan model yang terpilih dalam proses *model selection* ini.

Pergeseran Paradigma

Kemajuan teknologi komputasi mendorong berbagai perubahan dan perkembangan dalam analitik. Perkembangan tidak hanya terjadi dengan munculnya algoritma dan teknik baru, yang awalnya tidak mudah dan tidak murah dari sisi komputasi. Perkembangan juga terjadi pada paradigma penggunaan model akhir dalam melakukan prediksi.

Pada saat komputasi masih menjadi kendala besar dalam pemodelan, ada pemikiran bahwa algoritma yang diterapkan tinggal menggunakan salah satu saja dari yang tersedia. Pahlawannya untuk memperoleh model dari satu algoritma bisa jadi memerlukan waktu yang tidak sedikit. Dengan teknologi terkini, satu buah algoritma dapat menghasilkan sebuah model prediktif dalam waktu yang singkat apabila data-data yang diperlukan telah tersedia.



Gambar 1.
Mekanisme dasar
pendekatan ensemble
learning (Tsiaklis, 2015).

BUSINESS ANALYTIC

Kondisi ini kemudian memunculkan ide untuk melakukan prediksi tidak hanya didasarkan pada satu buah model (yang dianggap paling baik), namun melakukan prediksi dengan cara menggabungkan hasil prediksi dari banyak model. Paradigma ini yang dikenal sebagai *ensemble learning*. Theodoros Tsiaklis (2015) dalam buku yang dieditnya berjudul *Trends and Innovations in Marketing Information Systems* memuat bagaimana *ensemble learning* ini bekerja. Gambar 1 menyajikan secara ringkas sistem *ensemble* ini dipergunakan untuk melakukan prediksi.

Dari satu buah dataset dapat diperoleh banyak model prediktif baik menggunakan berbagai teknik yang berbeda maupun menggunakan algoritma yang sejenis. Setiap model selanjutnya menghasilkan prediksi yang dapat berbeda satu dengan yang lainnya. Pendekatan *ensemble learning* menggabungkan berbagai macam prediksi tersebut menjadi satu buah prediksi akhir. Teknik penggabungan yang banyak digunakan adalah *averaging* dan *majority vote*. Pada penerapan *majority vote* untuk *credit scoring* misalnya, keputusan apakah individu yang mengajukan aplikasi pinjaman akan ditolak atau diterima aplikasinya didasarkan pada suara terbanyak dari hasil prediksi macet-lancar dari banyak model.

Secara umum *ensemble learning* terbagi menjadi dua kelompok yaitu *hybrid ensemble* dan *non-hybrid ensemble*. Yang disebut *hybrid ensemble* adalah jika model-model yang nanti digabungkan prediksinya merupakan model-model yang dihasilkan dari berbagai jenis algoritma berbeda. Sementara *non-hybrid ensemble* menggunakan model-model yang diperoleh dari algoritma sejenis.

Ensemble Learning, Pilihan yang Tepat

Kenyataan bahwa pendekatan *ensemble learning* mampu memberikan solusi prediksi yang lebih akurat daripada model-model tunggal dapat ditemui dari berbagai paper di jurnal ilmiah. Teknik-teknik *ensemble* yang mengadopsi variasi dari pendekatan *random forest* dan *boosting* mampu memberikan prediksi dengan akurasi yang sangat baik. *Random forest* bekerja dengan membuat model-model penyusun *ensemble* sedemikian rupa sehingga berbagai kemungkinan dapat terakomodir secara maksimal, sedangkan *boosting* bekerja secara *iterative* sehingga kasus-kasus yang tidak mudah diprediksi menjadi bukan masalah lagi.

Kemampuan pendekatan *ensemble* ini tidak hanya terbatas pada berbagai paper ilmiah, namun juga dapat dilihat pada penyelesaian kasus-kasus aplikasi seperti yang dapat dilihat pada kompetisi *data science Kaggle* (<https://www.kaggle.com/>). Kompetisi ini terbuka bagi pejat *data science* dan *data mining* untuk memberikan solusi prediktif dari kasus-kasus yang disampaikan oleh banyak perusahaan besar berkala internasional.

Setiap tim atau individu dipisahkan mengembangkan solusi dan menyajikan prediksinya untuk kemudian dinilai. Mereka yang memberikan prediksi dengan akurasi yang paling tinggi yang dinyatakan sebagai pemenang. Peringkat tiga besar dalam lima tahun terakhir dari kompetisi ini didominasi oleh mereka yang menggunakan pendekatan *ensemble* yang digabungkan dengan berbagai macam algoritma dasar.

Berdasarkan apa yang berkenaan saat ini, pendekatan *ensemble* dalam pemodelan prediktif menjadi pilihan tepat bagi mereka yang berupaya memperoleh prediksi yang memuaskan dengan cara yang sangat mudah untuk dikerjakan. Hal senada juga telah dikemukakan oleh Mu Zhu (University of Waterloo) pada Jurnal *The American Statistician* pada tahun 2008. ■

Outline

- Pengantar
- Bagging, Random Forest
- Boosting
- Lain-lain [optional]:
 - Ensemble untuk pemodelan klasifikasi pada data tidak seimbang

Prinsip Dasar



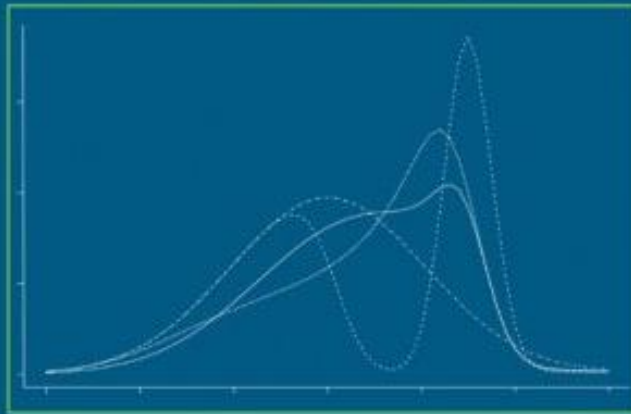
**single
expert**

vs



**a team of
experts**

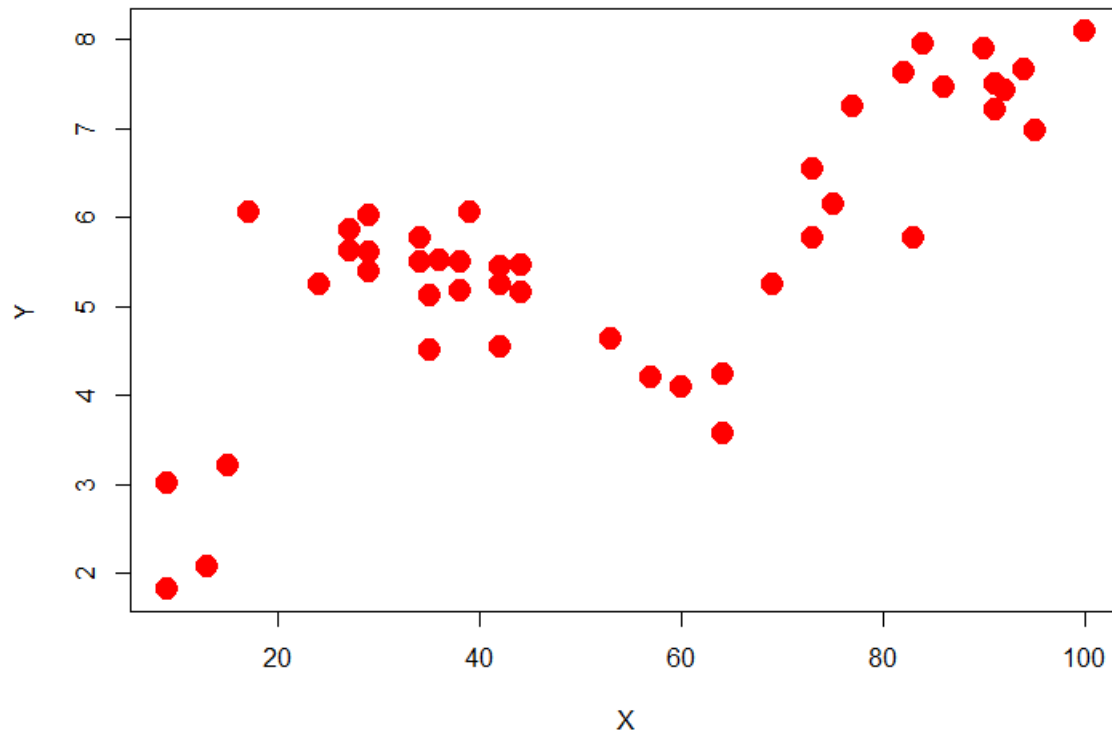
Cambridge Series in Statistical
and Probabilistic Mathematics



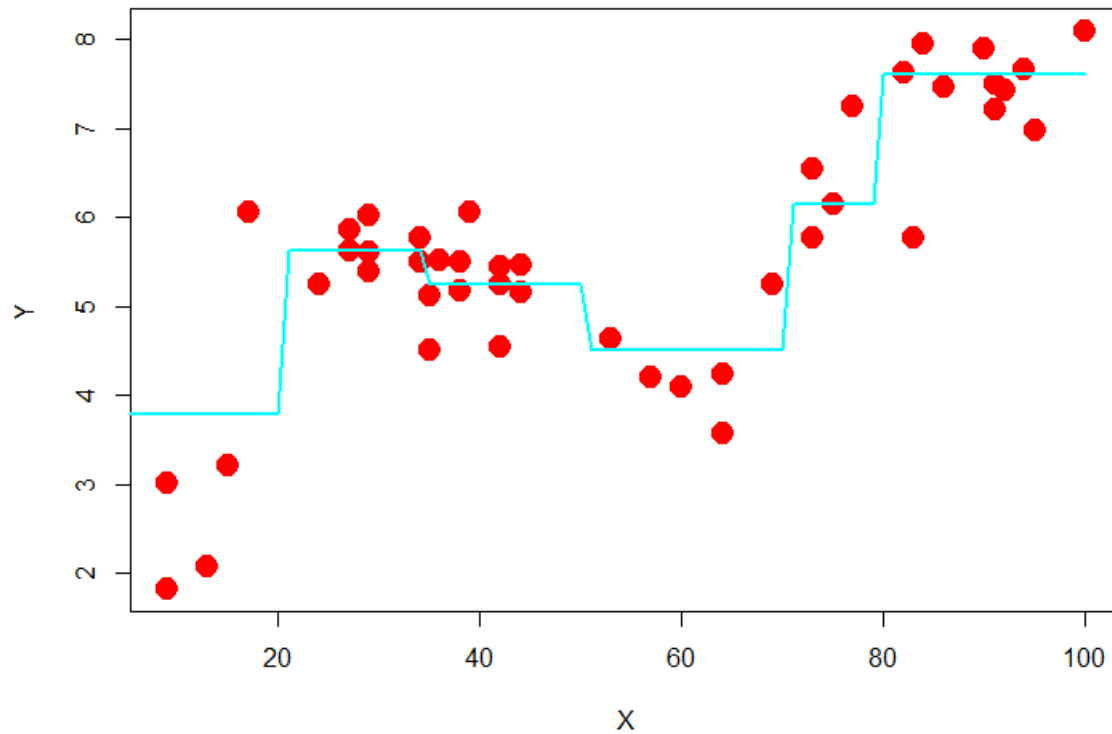
Model Selection and Model Averaging

Gerda Claeskens and Nils Lid Hjort

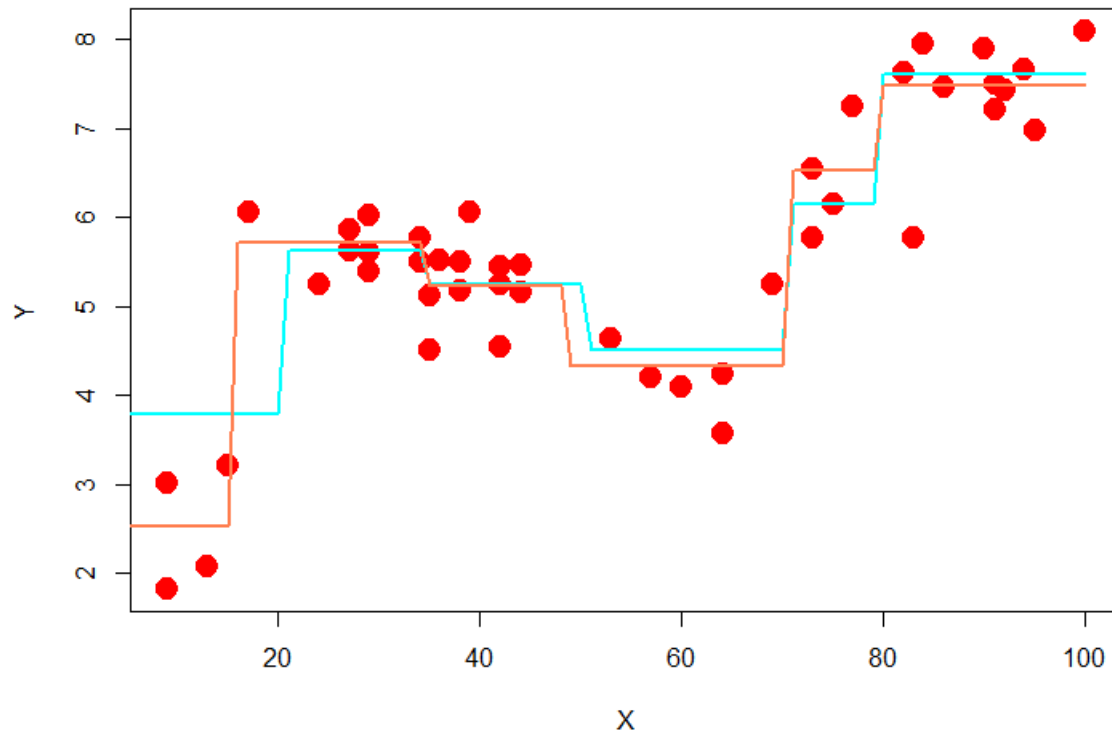
Motivating Example #1



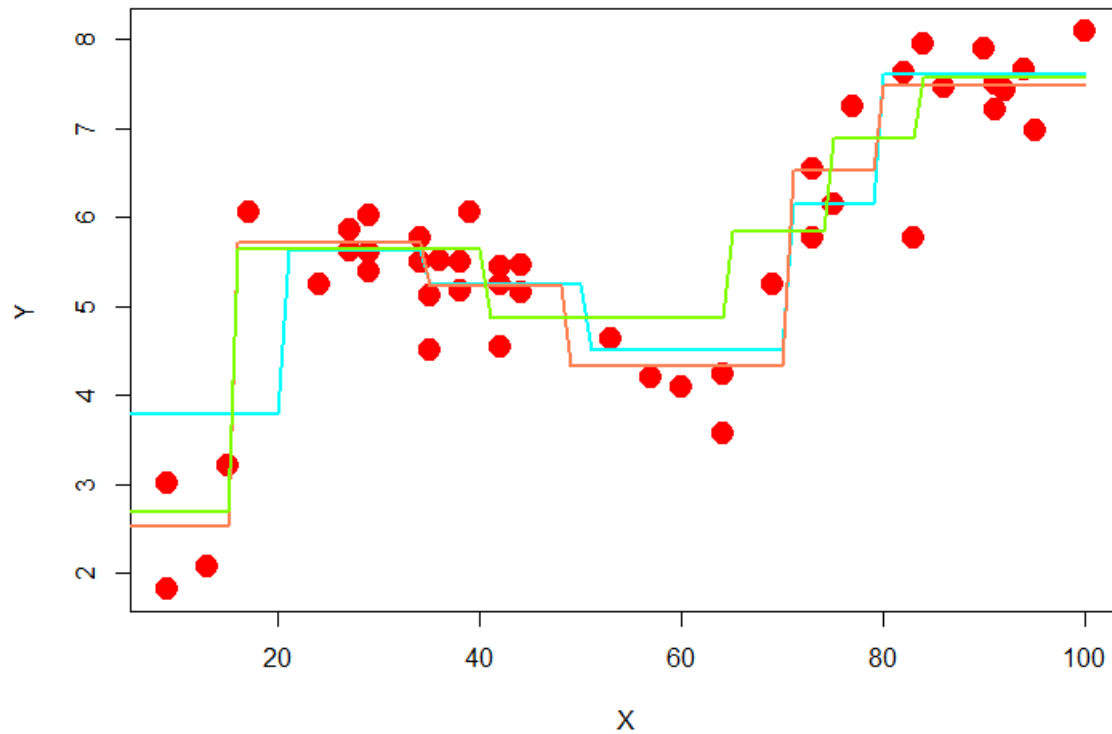
Motivating Example #1



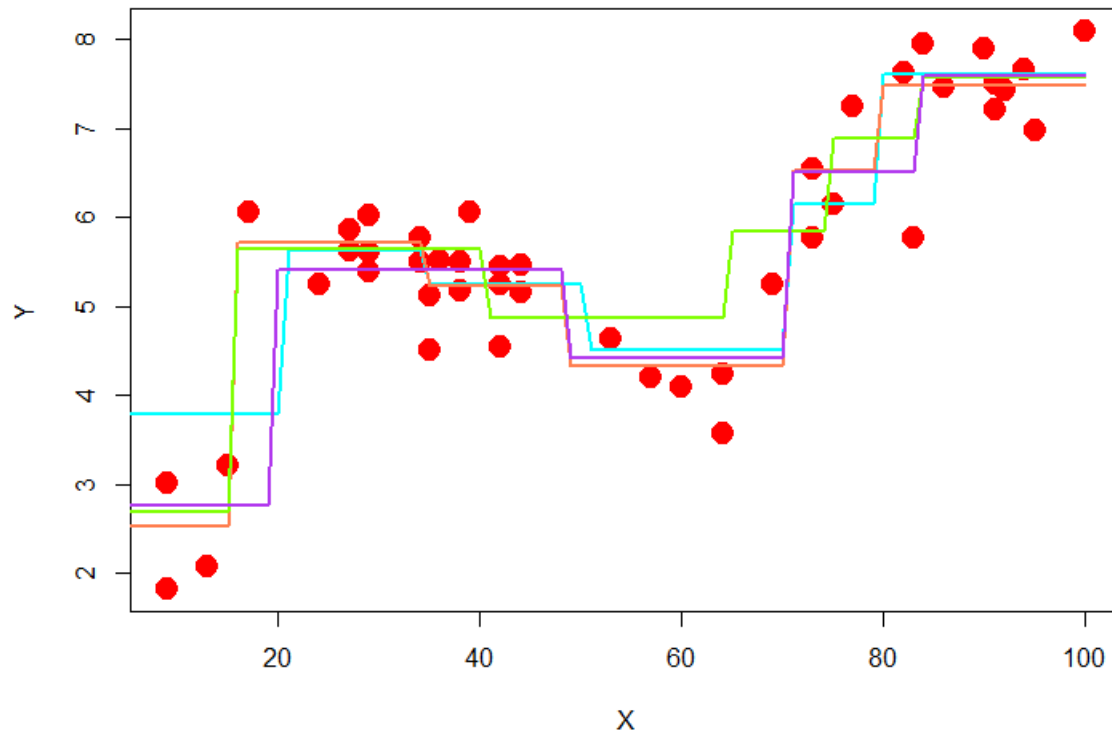
Motivating Example #1



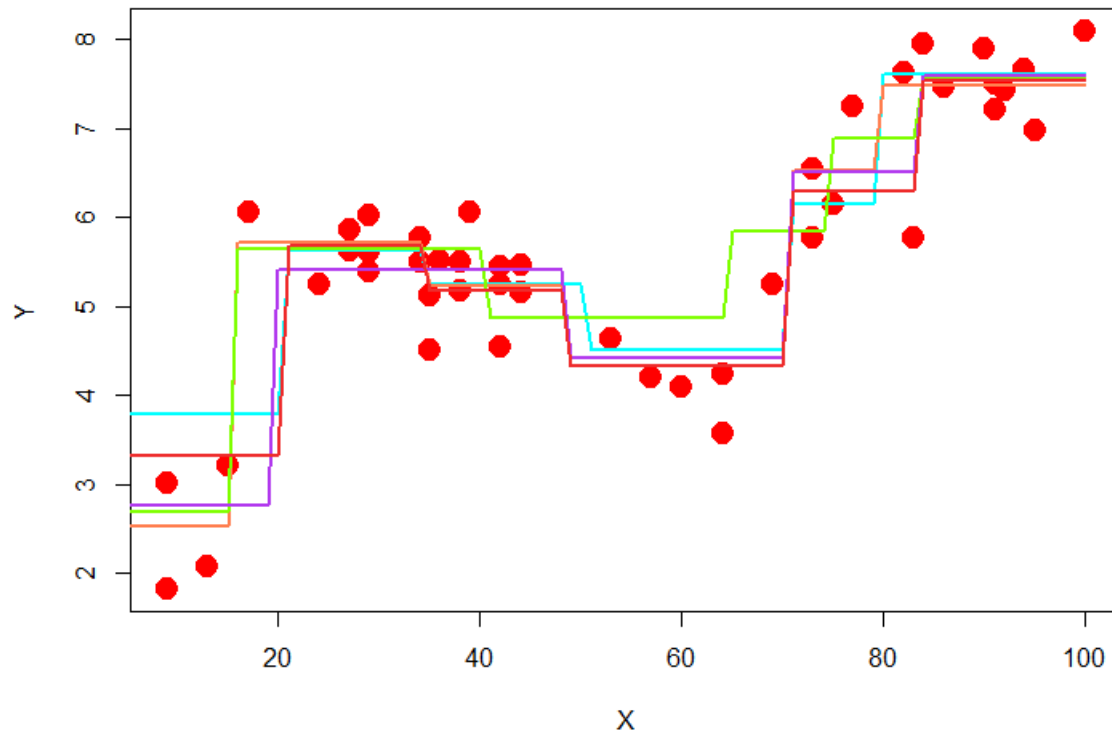
Motivating Example #1



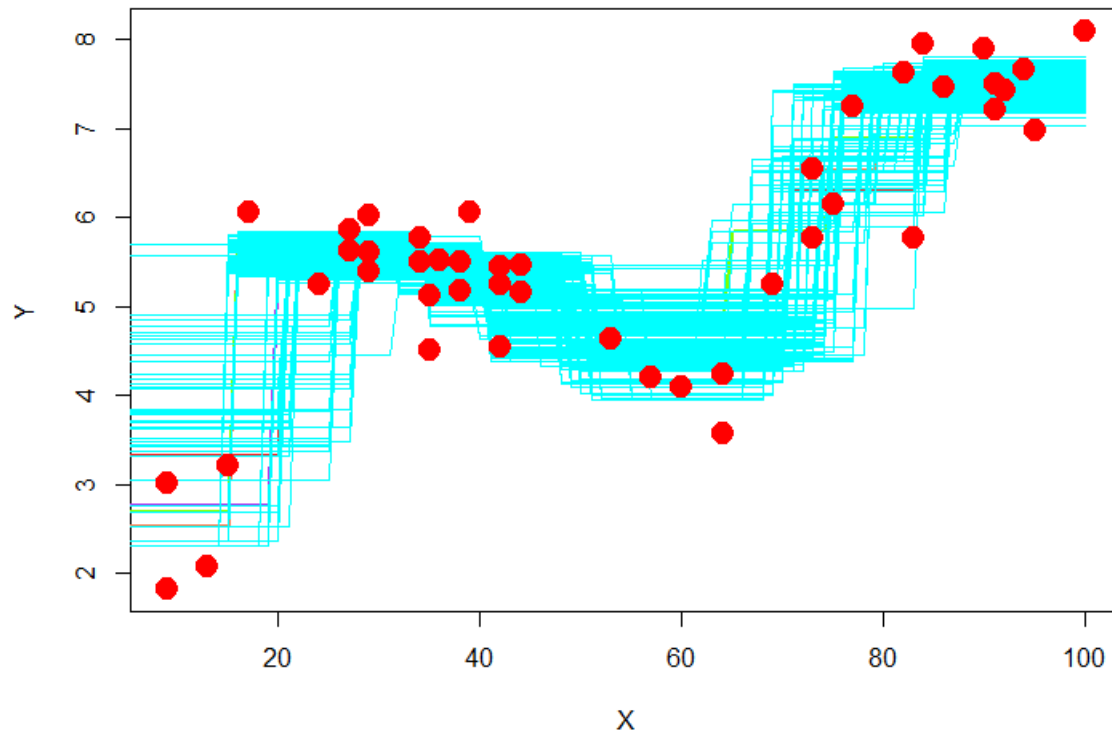
Motivating Example #1



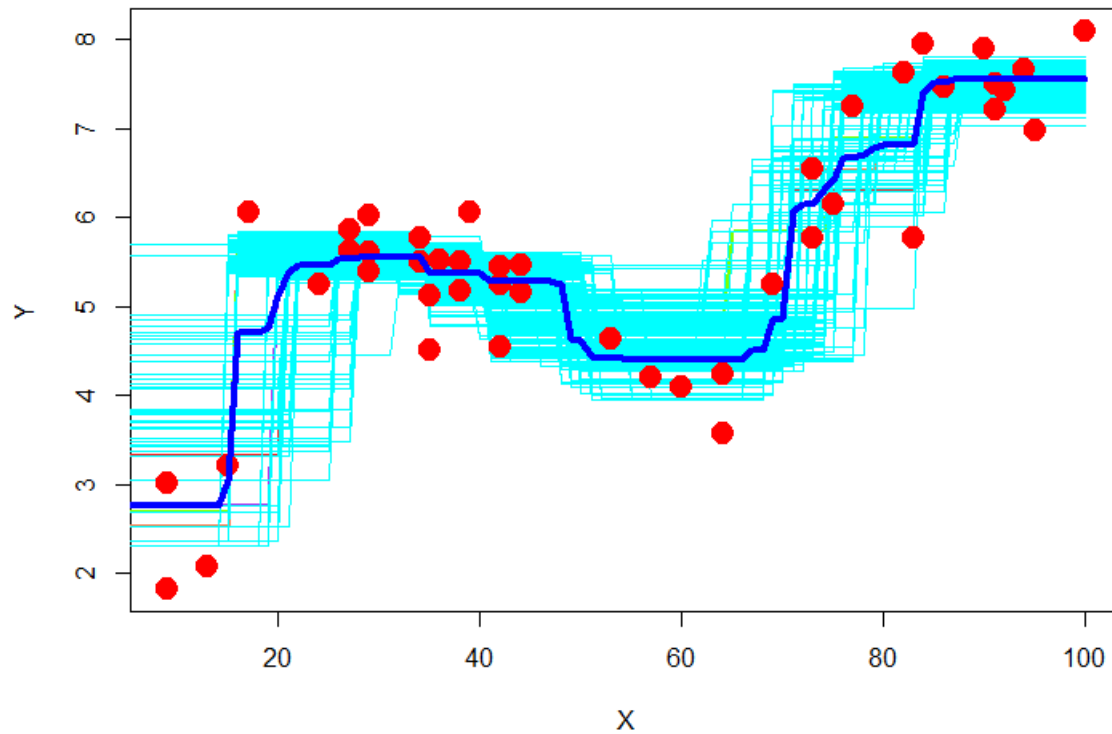
Motivating Example #1



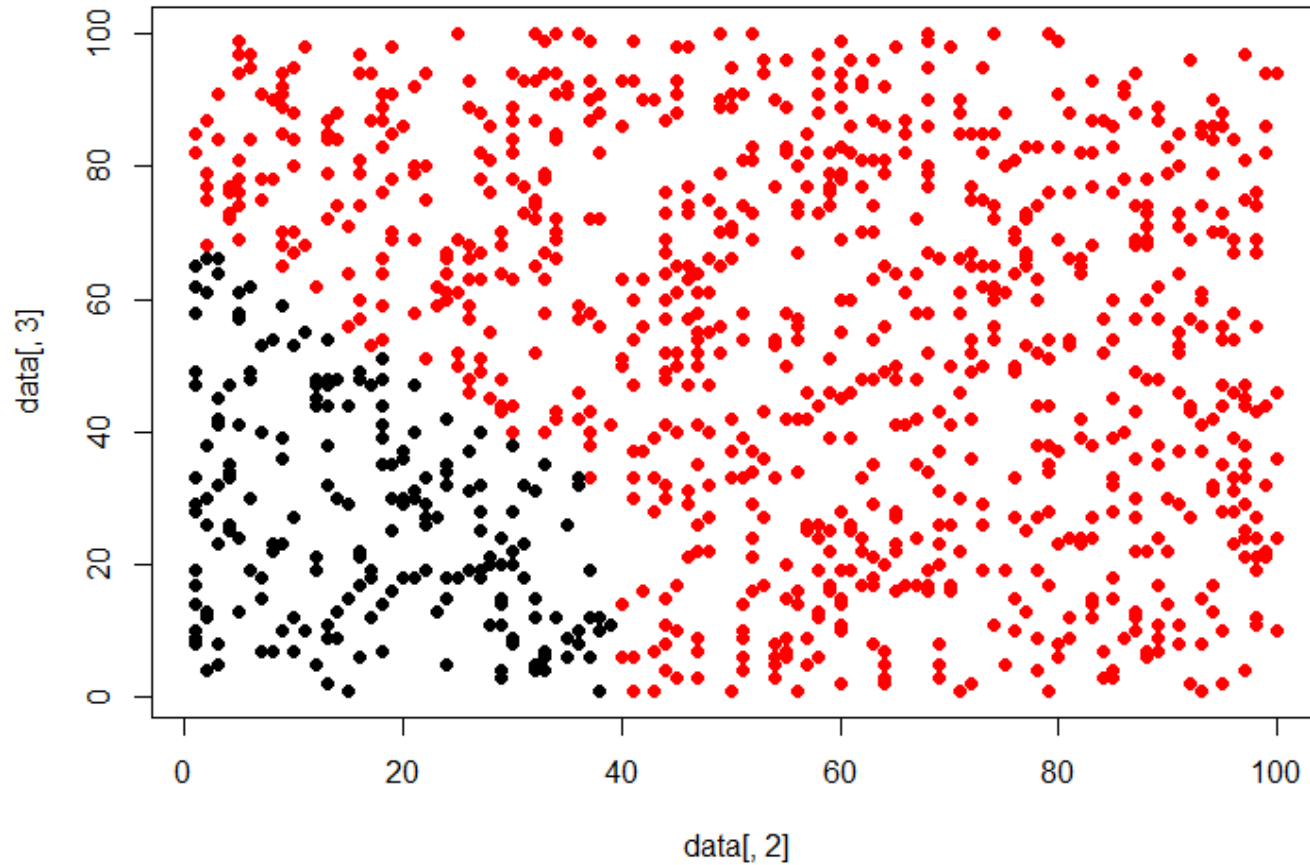
Motivating Example #1



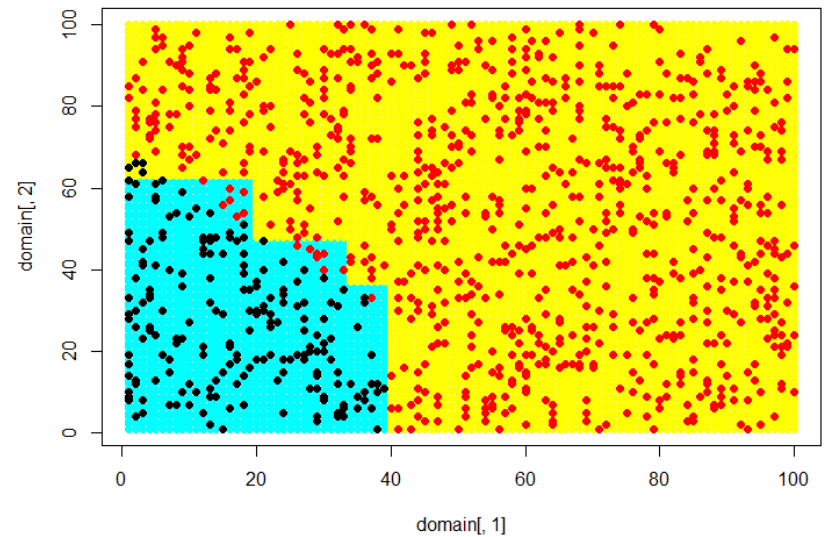
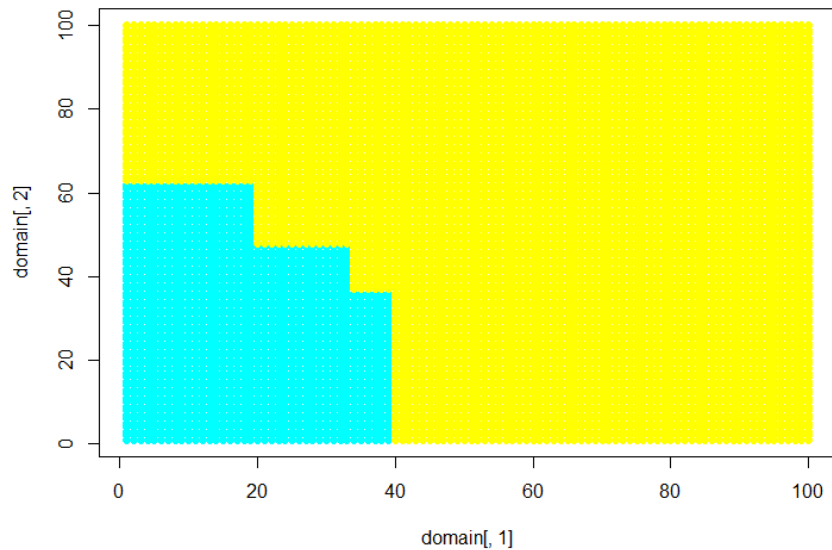
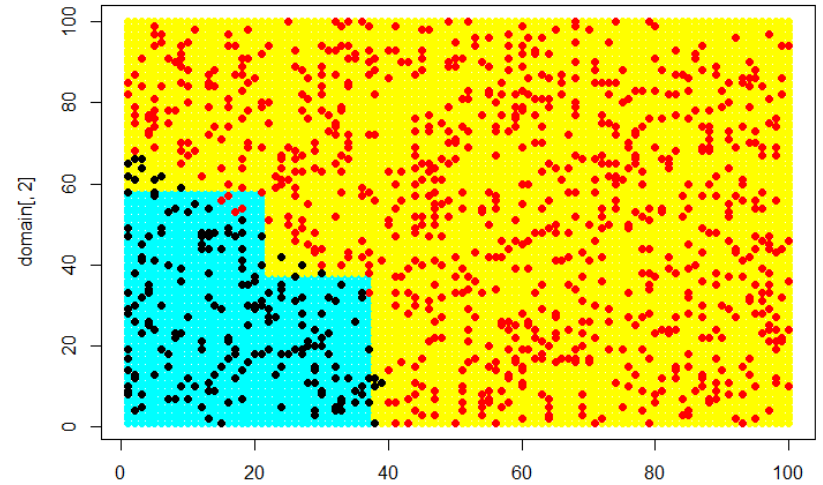
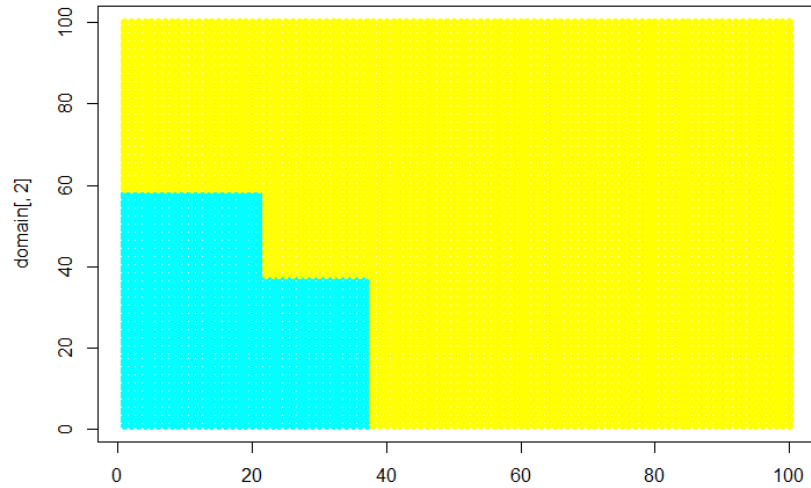
Motivating Example #1



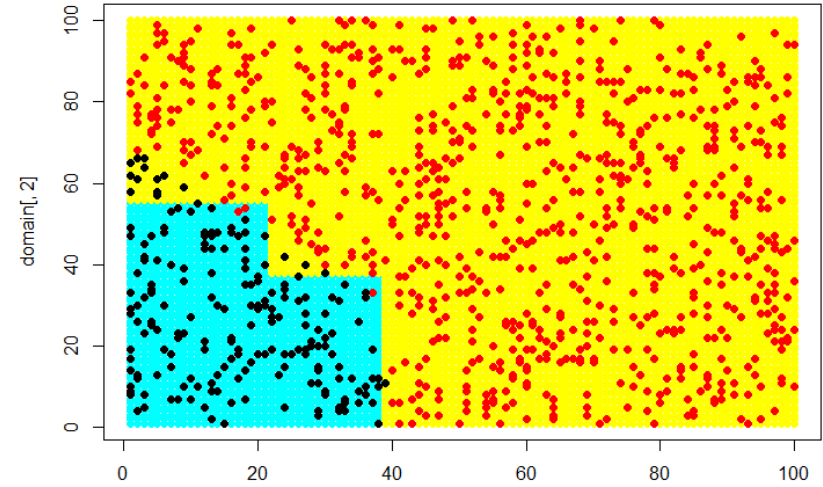
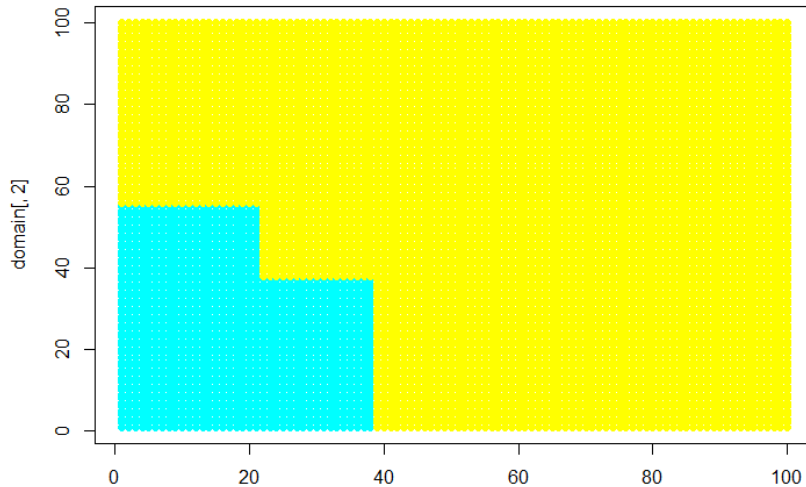
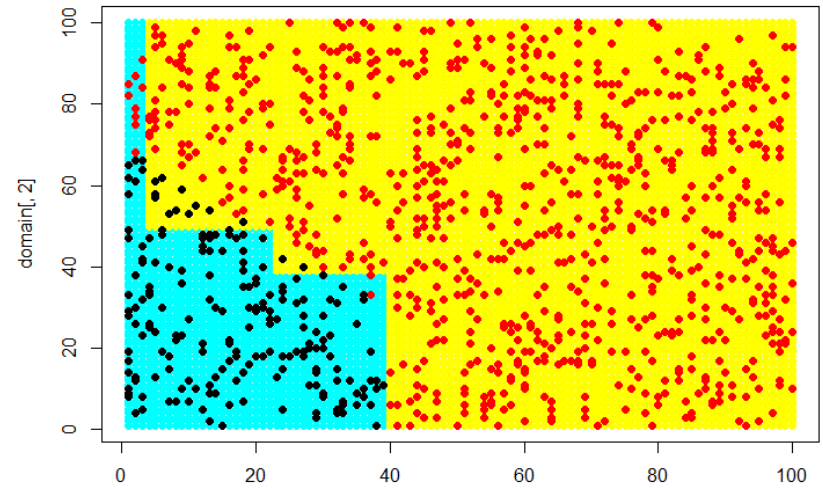
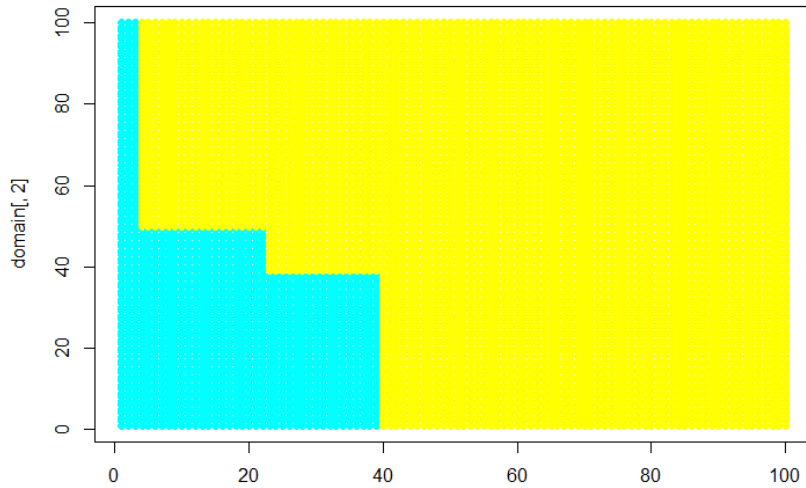
Motivating Example #2



Motivating Example #2



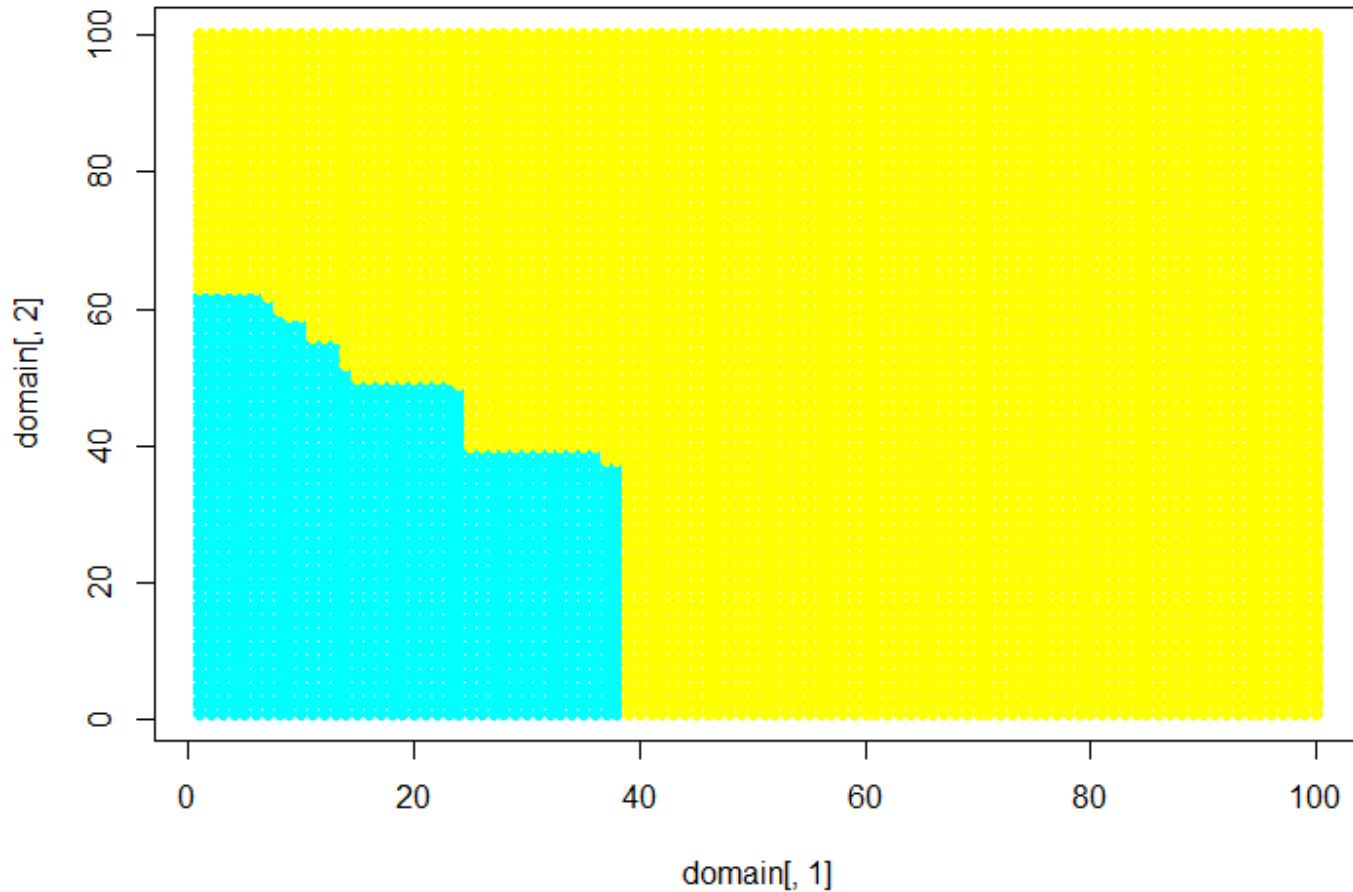
Motivating Example #2



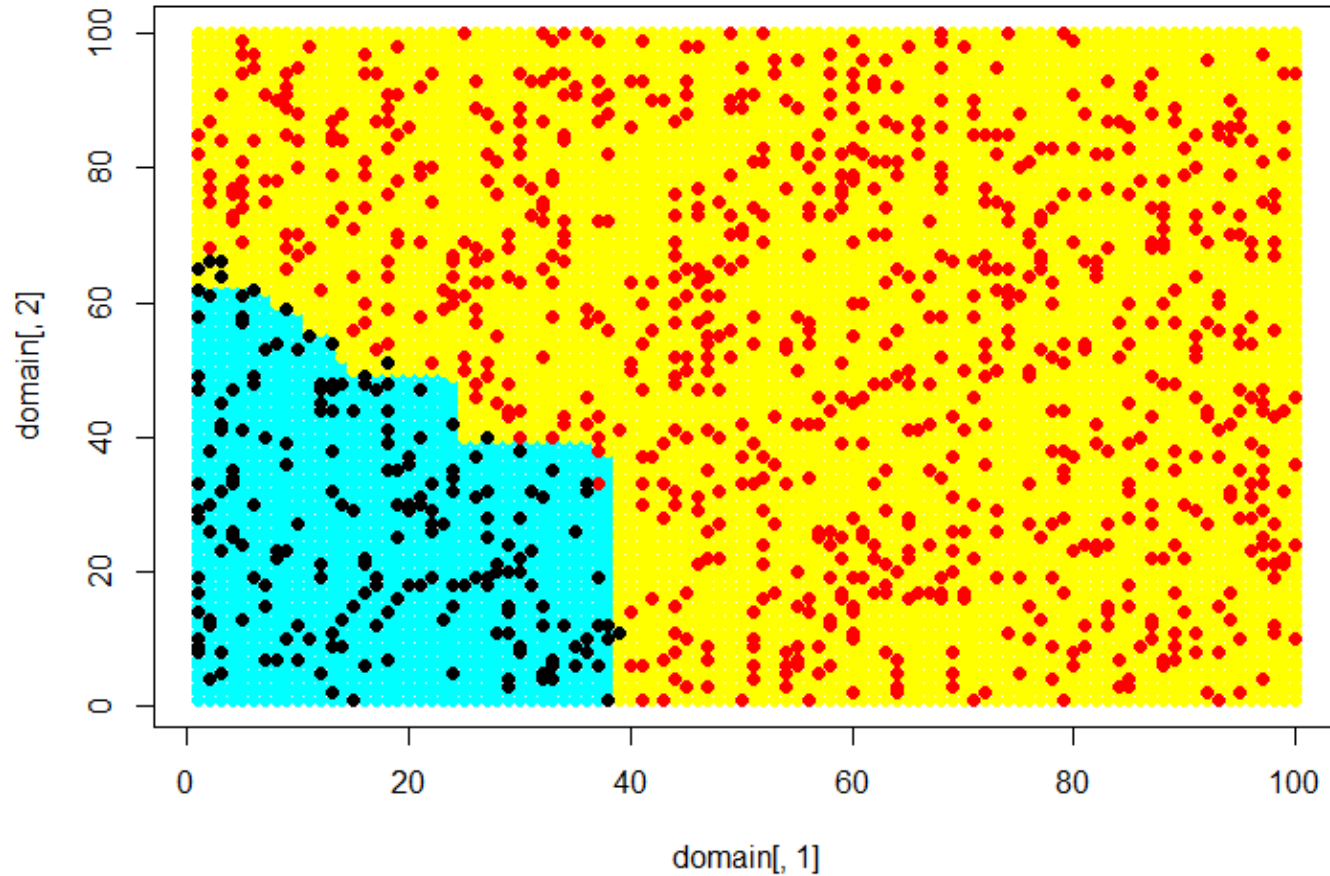
domain[1]

domain[1]

Motivating Example #2



Motivating Example #2



Ide Dasar

- Andaikan ingin diprediksi status kolektabilitas nasabah berdasarkan variabel prediktor berikut

age	Age in years
ed	Level of education
employ	Years with current employer
address	Years at current address
income	Household income in thousands
debtinc	Debt to income ratio (x100)
creddebt	Credit card debt in thousands
othdebt	Other debt in thousands

- Model prediktif yang mungkin digunakan: binary logistic regression (BLR), discriminant analysis (DA), dll

Ide Dasar

Nasabah	BLR	DA	Dugaan
1	0	1	0
2	1	0	1
3	0	0	0
4	0	0	0
5	0	0	0
6	1	1	1
7	1	0	0
8	0	1	0
...			

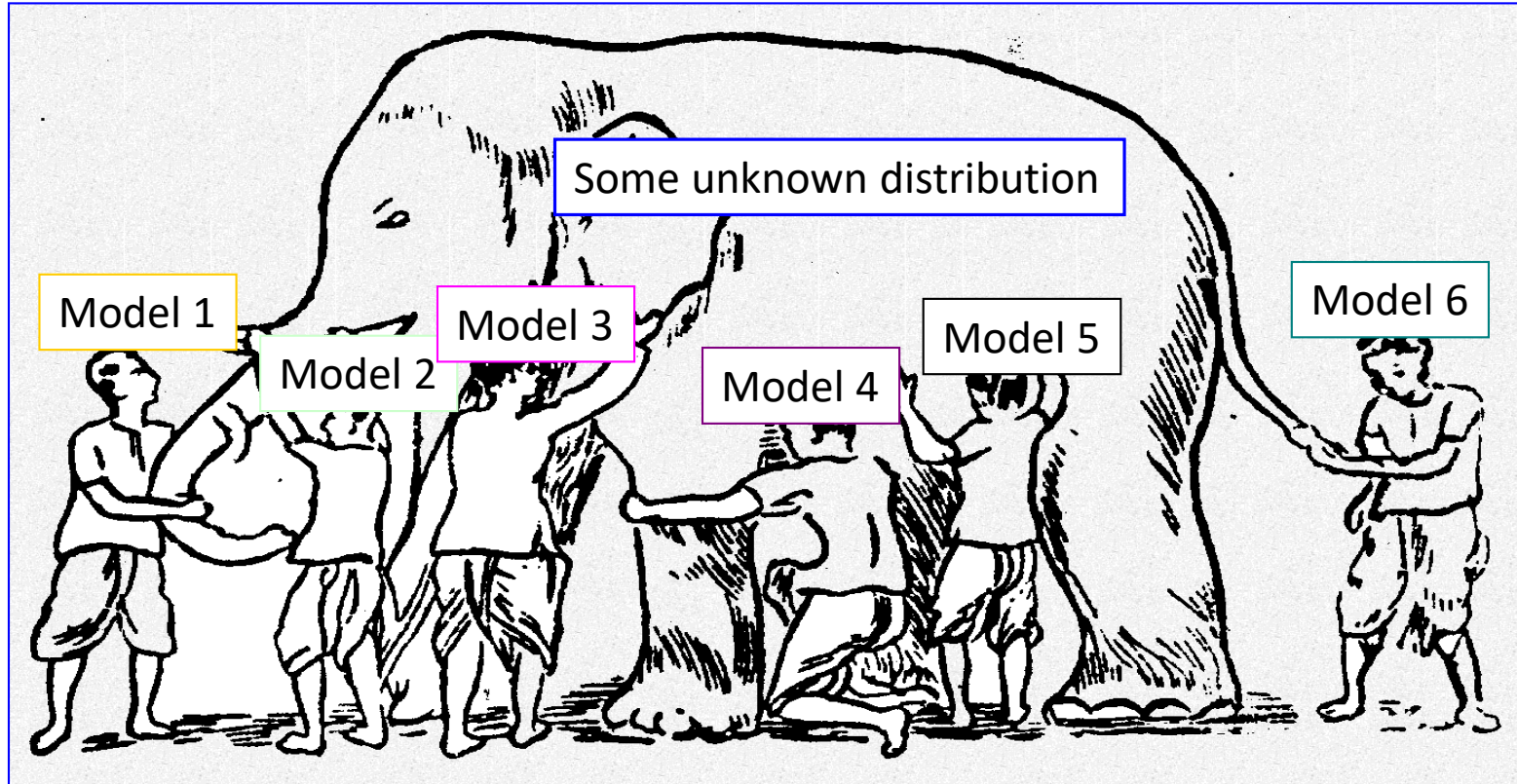
note: 1 → default; 0 → not default

- Setelah pemodelan, diperoleh akurasi klasifikasi:
 - BLR: 80%
 - DA: 78%
 - Metode mana yang akan dipilih?
 - Bisakah kita gabungkan kedua metode itu agar akurasi meningkat?
- Ensemble Approach

Ide Dasar

- Tidak tersedia algoritma yang selalu paling akurat
- Bangkitkan satu gugus base-learners yang kalau digabungkan bisa memberikan akurasi yang lebih tinggi
- Tiap base-learner dapat berbeda dalam hal:
 - Algoritma
 - Hyperparameter
 - Gugus data training
 - Subproblems

Why Ensemble Works?



Ensemble gives the global picture!

Why does it work?

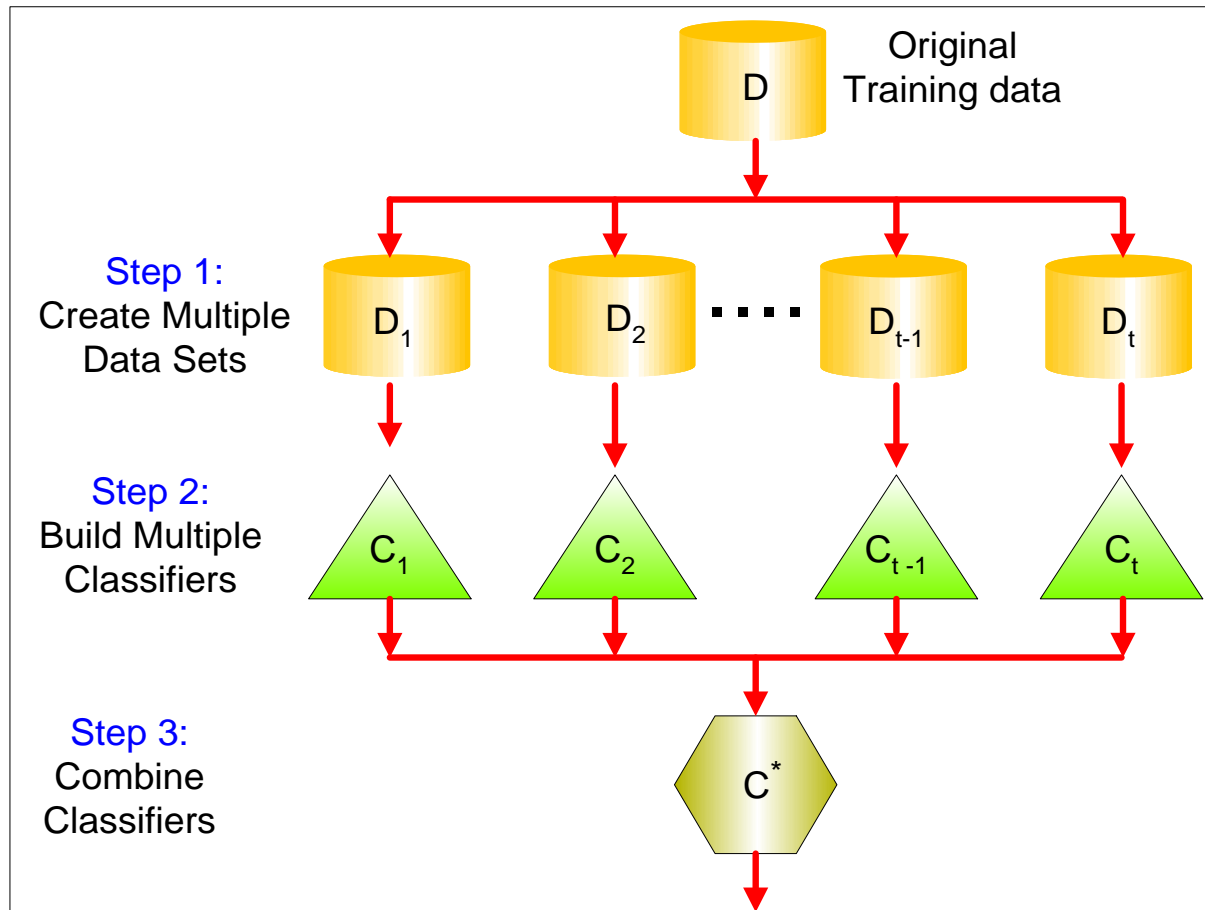
- Suppose there are 25 base classifiers
 - Each classifier has error rate, $\varepsilon = 0.35$
 - Assume classifiers are independent
 - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1 - \varepsilon)^{25-i} = 0.06$$

What is the Main Challenge for Developing Ensemble Models?

- The main challenge is **not** to obtain **highly accurate base models**, but rather to **obtain base models which make different kinds of errors**.
- High accuracies can be accomplished if **different base models misclassify different training examples**, even if the base classifier accuracy is low.

Ide Dasar



BAGGING

Pengantar

- Bagging, bootstrap + aggregating
- Breiman, L .1996. Bagging predictors. *Machine Learning*. 24 (2): 123–140.

Algorithm 5.6 Bagging algorithm.

- 1: Let k be the number of bootstrap samples.
 - 2: **for** $i = 1$ to k **do**
 - 3: Create a bootstrap sample of size N , D_i .
 - 4: Train a base classifier C_i on the bootstrap sample D_i .
 - 5: **end for**
 - 6: $C^*(x) = \operatorname{argmax}_y \sum_i \delta(C_i(x) = y)$.
 $\{\delta(\cdot) = 1 \text{ if its argument is true and } 0 \text{ otherwise}\}.$
-

Bagging

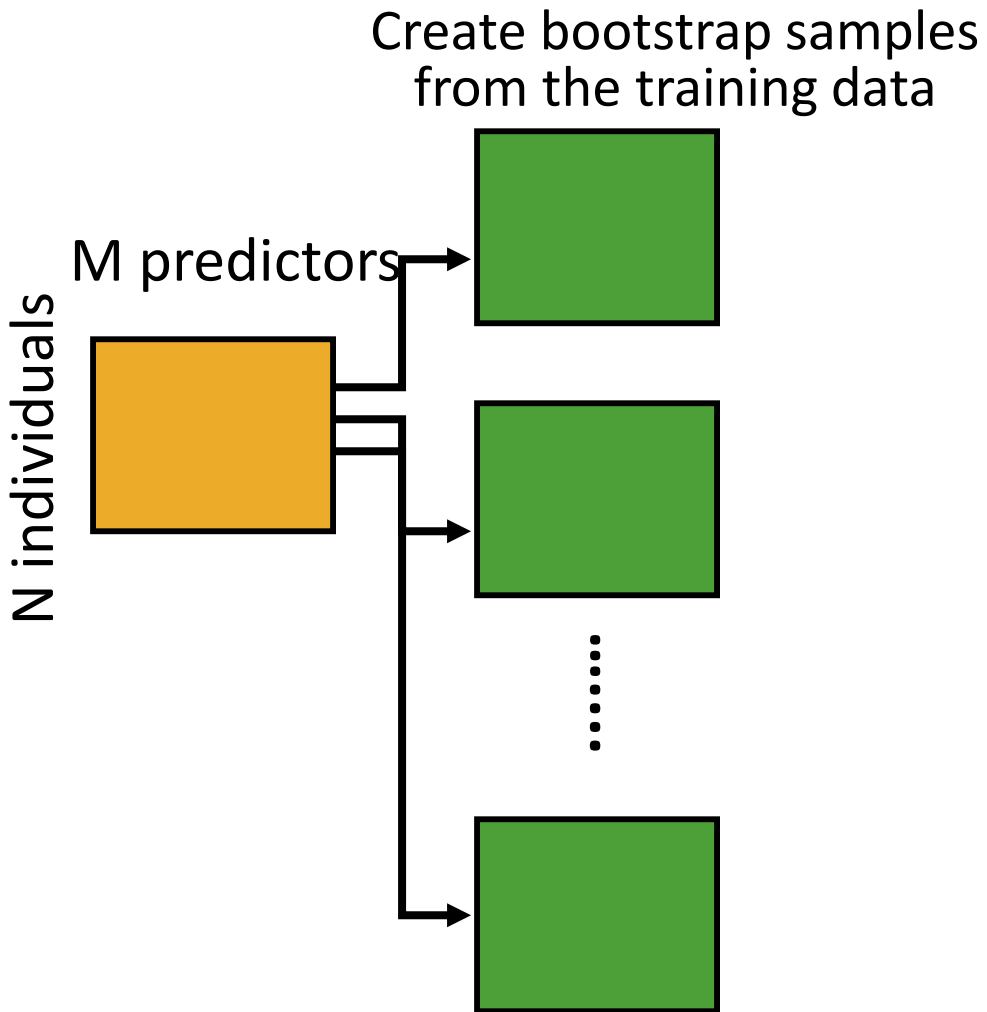
Training Data

N individuals

M predictors

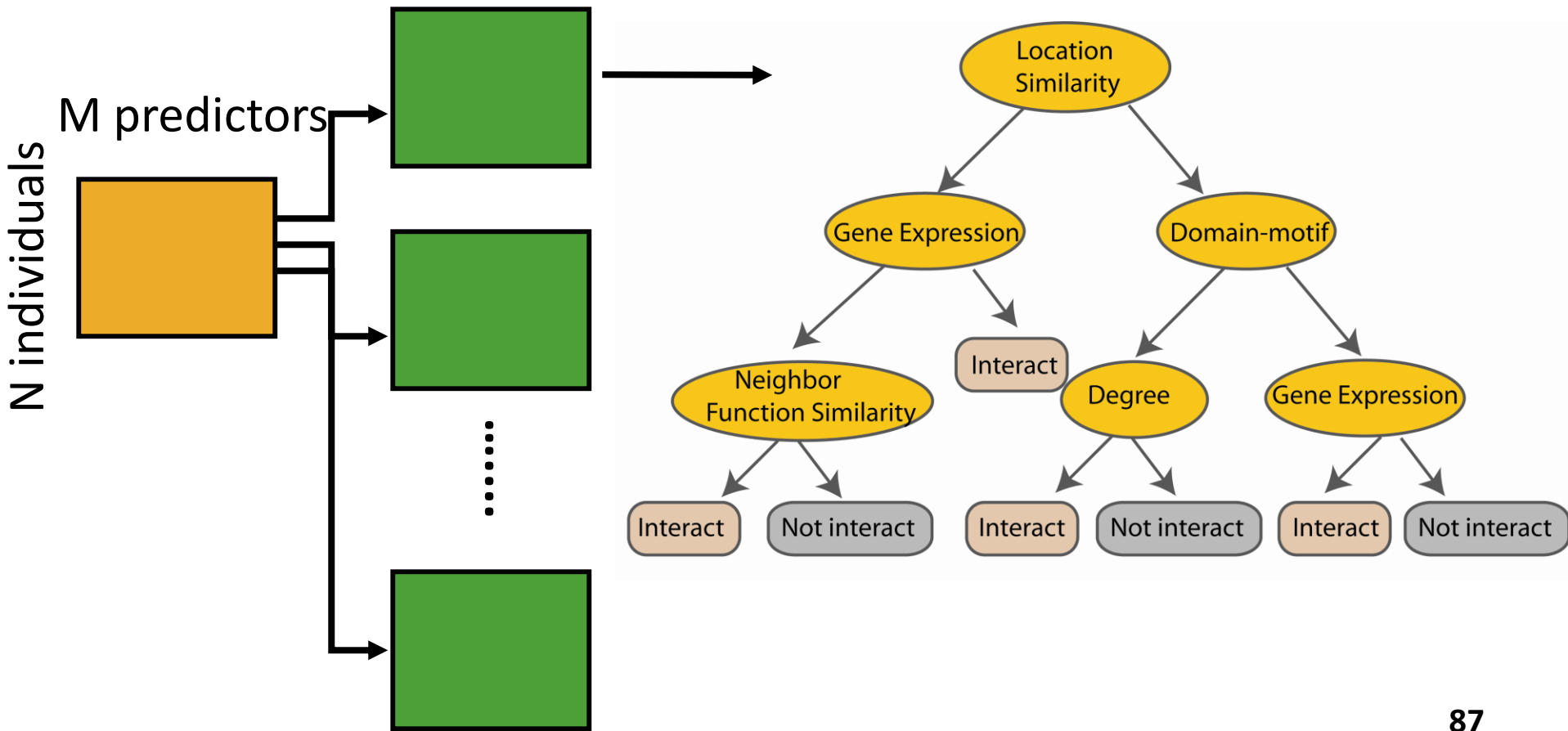


Bagging



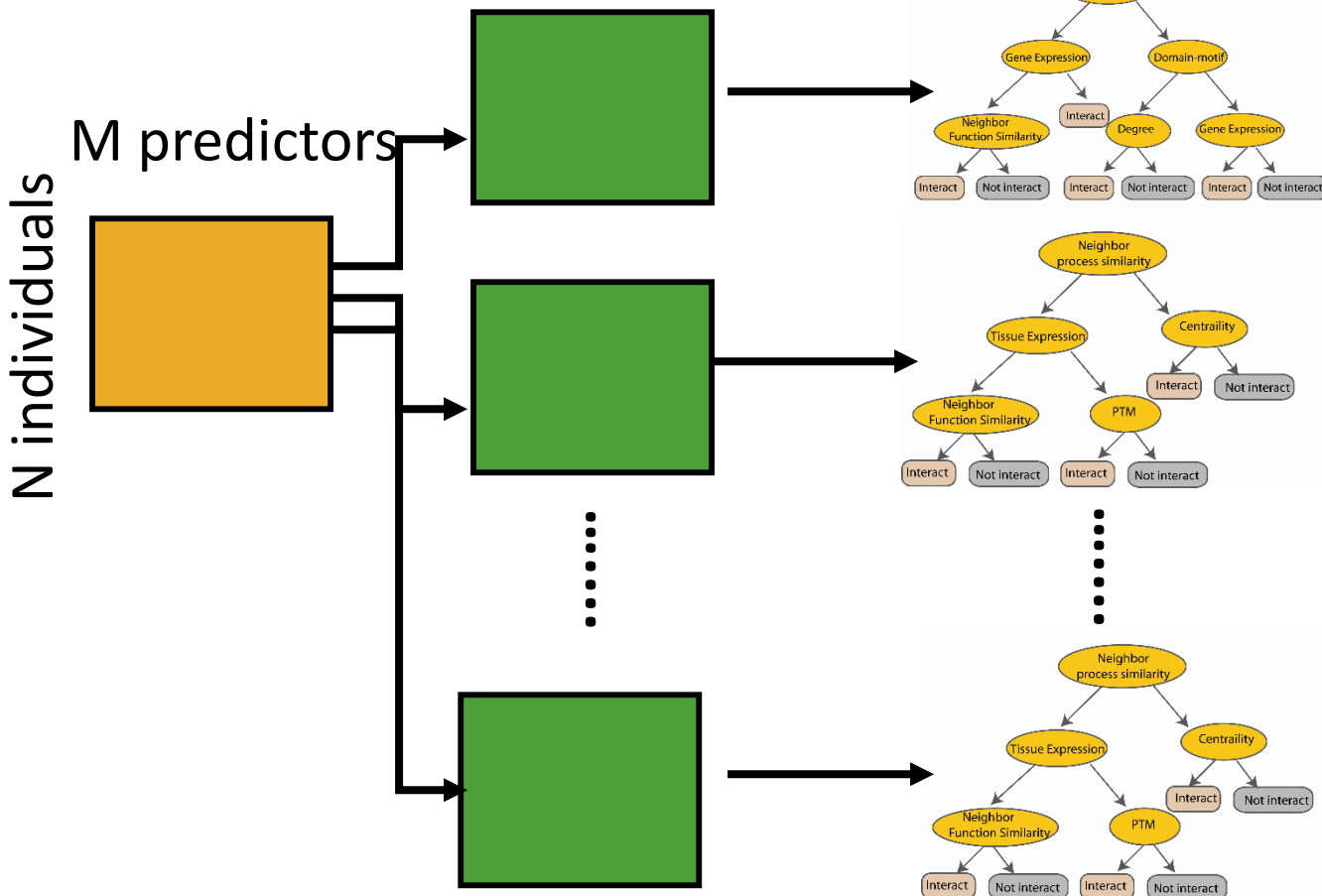
Bagging

Construct a decision tree



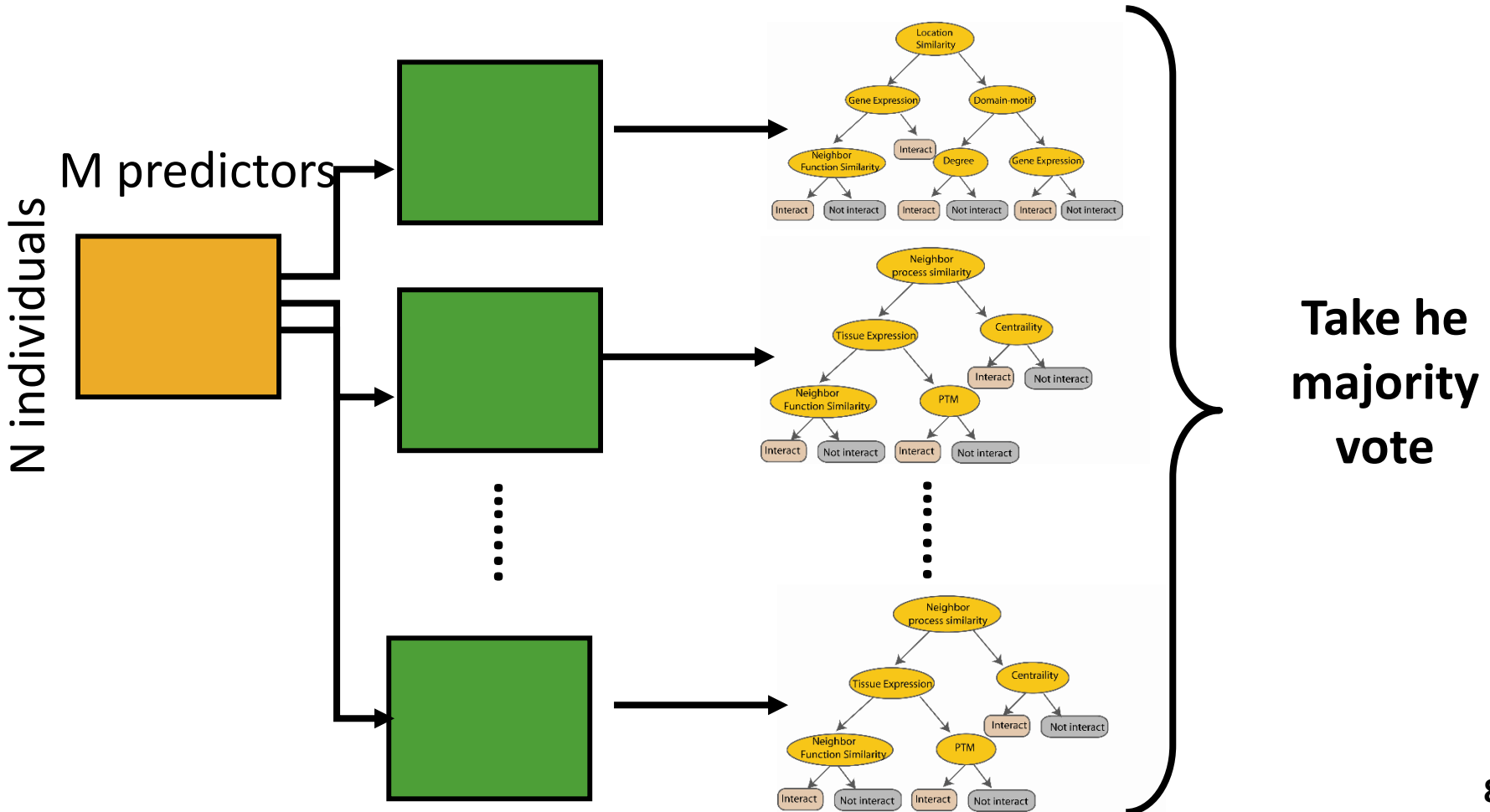
Bagging

Create decision tree
from each bootstrap sample



Bagging

Create decision tree
from each bootstrap sample

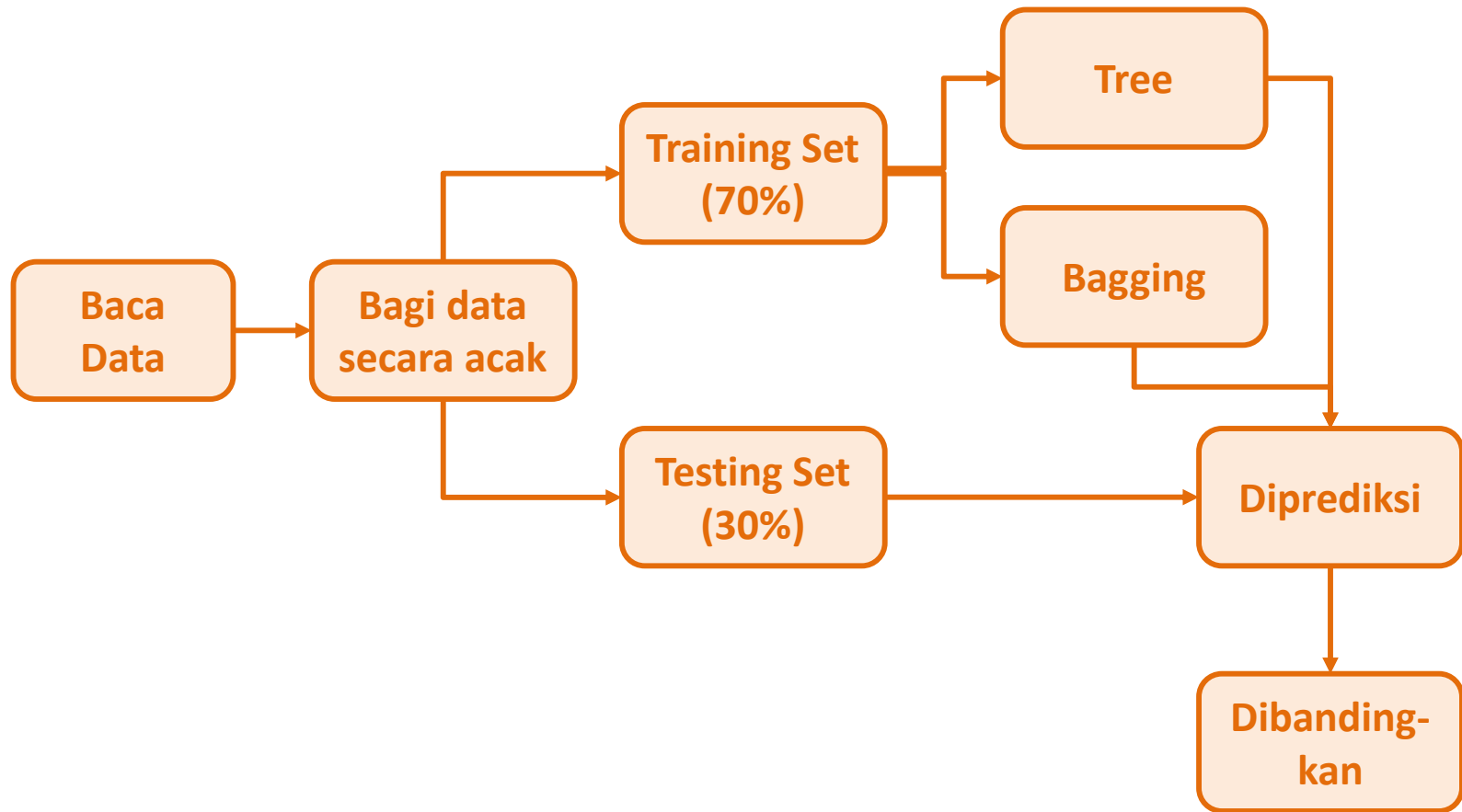


Mari kita coba di R

- Bandingkan performa tree (pohon tunggal) dengan bagging....

- Data diadopsi dari <https://archive.ics.uci.edu/ml/datasets/ionosphere>
- Data dan program bisa didownload pada <https://github.com/bagusco/tadulako>
- 34 prediktor

Alur Analisis



Menyiapkan data

```
#membaca data
```

```
alamat <- "D:/"
```

```
ion <- read.csv(paste0(alamat,"Ionosphere_ok.csv"))[, -1]
```

```
#membagi dataset menjadi dua dataset
```

```
library(caret)
```

```
set.seed(100)
```

```
idx <- createDataPartition(ion$Class, p=0.7, list=FALSE)
```

```
train <- ion[idx,]
```

```
test <- ion[-idx,]
```

Tree dan prediksinya

```
#membuat pohon klasifikasi dan memprediksi data testing  
library(rpart)  
mod.tree <- rpart(Class~., data=train, method="class")  
  
prob <- predict(mod.tree, test)[,2]  
  
pred.tree <- as.factor(ifelse(prob>.5,"good","bad"))
```

Bagging dan prediksinya

#bagging dan memprediksi data testing

```
k<-50
```

```
prediksi <- matrix(NA,nrow(test),k)
```

```
for(i in 1:k) {
```

```
  resample <- sample(1:nrow(train), replace=TRUE)
```

```
  contoh.boot <- train[resample,]
```

```
  tree <- rpart(Class~., data=contoh.boot, method="class")
```

```
  prob <- predict(tree, test)[,2]
```

```
  prediksi[,i] <-ifelse(prob<0.5, 0, 1)
```

```
}
```

```
vote1 <- apply(prediksi,1,sum)
```

```
pred.bag <- as.factor(ifelse(vote1 < k/2, "bad", "good"))
```

Membandingkan antara kelas prediksi dan aktual

```
library(caret)
kinerja.tree <-
confusionMatrix(pred.tree, test$Class, positive = "good")

kinerja.bagging <-
confusionMatrix(pred.bag, test$Class, positive = "good")

kinerja.tree
kinerja.bagging
```



```
> kinerja.tree
```

Confusion Matrix and Statistics

	Reference	
Prediction	bad	good
bad	24	2
good	13	65

Accuracy : 0.8558

95% CI : (0.7733, 0.917)

No Information Rate : 0.6442

P-Value [Acc > NIR] : 1.262e-06

Kappa : 0.6629

McNemar's Test P-Value : 0.009823

Sensitivity : 0.9701

Specificity : 0.6486

Pos Pred Value : 0.8333

Neg Pred Value : 0.9231

Prevalence : 0.6442

Detection Rate : 0.6250

Detection Prevalence : 0.7500

Balanced Accuracy : 0.8094

'Positive' Class : good

```
> kinerja.bagging
```

Confusion Matrix and Statistics

	Reference	
Prediction	bad	good
bad	29	2
good	8	65

Accuracy : 0.9038

95% CI : (0.8303, 0.9529)

No Information Rate : 0.6442

P-Value [Acc > NIR] : 1.168e-09

Kappa : 0.7823

McNemar's Test P-Value : 0.1138

Sensitivity : 0.9701

Specificity : 0.7838

Pos Pred Value : 0.8904

Neg Pred Value : 0.9355

Prevalence : 0.6442

Detection Rate : 0.6250

Detection Prevalence : 0.7019

Balanced Accuracy : 0.8770

'Positive' Class : good

RANDOM FOREST

Prinsip Dasar

- Breiman L (2001). "Random Forests". Machine Learning. 45 (1): 5–32
- Prinsipnya serupa dengan bagging yaitu bekerja dengan subset dari data.
- Perbedaan dengan bagging, RF juga melakukan subset terhadap variabel input, tidak hanya subset pengamatan.

Random Forest

- Pohon yang dihasilkan akan memiliki diversity yang lebih tinggi dibandingkan bagging.
 - ada yang pendek ada yang tinggi, ada yang rindang ada yang menjulang
- Dalam beberapa studi empirik, hasilnya lebih baik dibandingkan bagging.
- Pengambilan kesimpulan ditentukan berdasarkan majority vote.

Algoritma

For $b = 1$ to B :

(a) Draw a bootstrap sample Z^* of size N from the training data.

(b) Grow a random-forest tree to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

i. Select m variables at random from the p variables.

ii. Pick the best variable/split-point among the m .

iii. Split the node into two daughter nodes.

Output the ensemble of trees.

To make a prediction at a new point x we do:

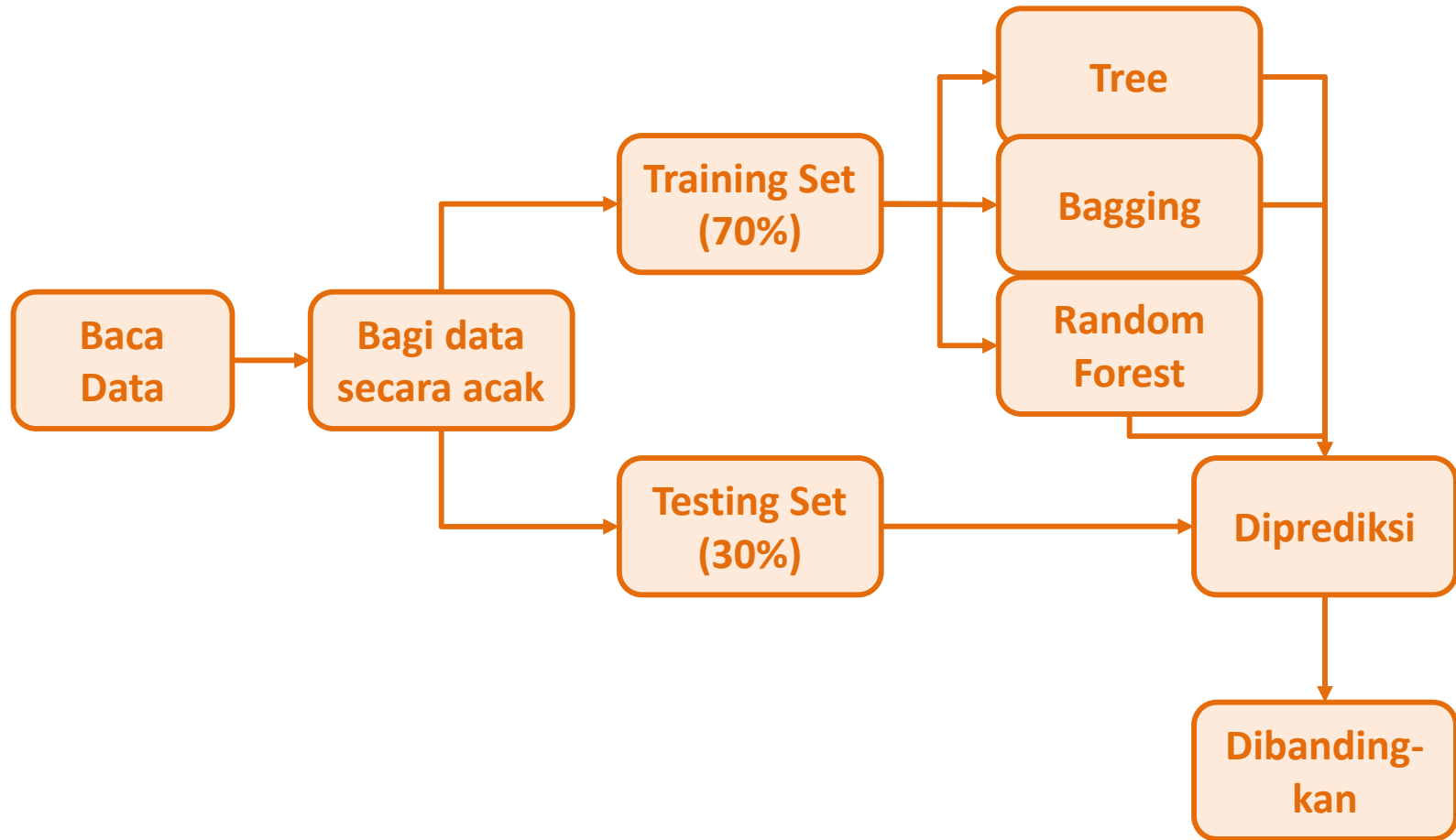
For regression: average the results

For classification: majority vote

Mari kita coba di R

- Bandingkan performa tree (pohon tunggal) dengan bagging dan random forest....

Alur Analisis



#pemodelan random forest dan memprediksi data testing

```
library(randomForest)
```

```
model.forest <- randomForest(Class~.,data=train,  
importance=TRUE, ntree=200, mtry=3)
```

```
pred.rf <- predict(model.forest, test)
```

```
kinerja.rf <- confusionMatrix(pred.rf, test$class,  
positive = "good")
```

```
kinerja.rf
```



```
> kinerja.rf
```

Confusion Matrix and Statistics

	Reference	
Prediction	bad	good
bad	32	3
good	5	64

Accuracy : 0.9231

95% CI : (0.854, 0.9662)

No Information Rate : 0.6442

P-Value [Acc > NIR] : 3.604e-11

Kappa : 0.8301

McNemar's Test P-Value : 0.7237

Sensitivity : 0.9552

Specificity : 0.8649

Pos Pred Value : 0.9275

Neg Pred Value : 0.9143

Prevalence : 0.6442

Detection Rate : 0.6154

Detection Prevalence : 0.6635

Balanced Accuracy : 0.9100

'Positive' Class : good

BOOSTING

Prinsip Dasar

- Prosesnya iteratif
- Melihat kesalahan dari pengklasifikasi awal, dan kemudian membuat pengklasifikasi baru pada iterasi berikutnya yang focus pada amatan yang salah klasifikasi
- Model yang baru tergantung pada model sebelumnya
- Ide utama: memberi bobot lebih besar pada amatan yang “sulit diduga” (yaitu amatan yang salah klasifikasi pada iterasi sebelumnya)

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
 - Initially, all N records are assigned equal weights
 - Unlike bagging, weights may change at the end of a boosting round
 - Different implementations vary in terms of (1) how the weights of the training examples are updated and (2) how the predictions are combined

Algoritma

Initialization step: for each example x , set

$$D(x) = \frac{1}{N}, \text{ where } N \text{ is the number of examples}$$

Iteration step (for $t=1 \dots T$):

1. Find best weak classifier $h_t(x)$ using weights $D_t(x)$

2. Compute the error rate ε_t as

$$\varepsilon_t = \sum_{i=1}^N D(x_i) \cdot I[y_i \neq h_t(x_i)]$$

3. assign weight α_t to classifier $h_t(x)$ in the final hypothesis

$$\alpha_t = \log((1 - \varepsilon_t) / \varepsilon_t)$$

4. For each x_i , $D(x_i) = D(x_i) \cdot \exp(\alpha_t \cdot I[y_i \neq h_t(x_i)])$

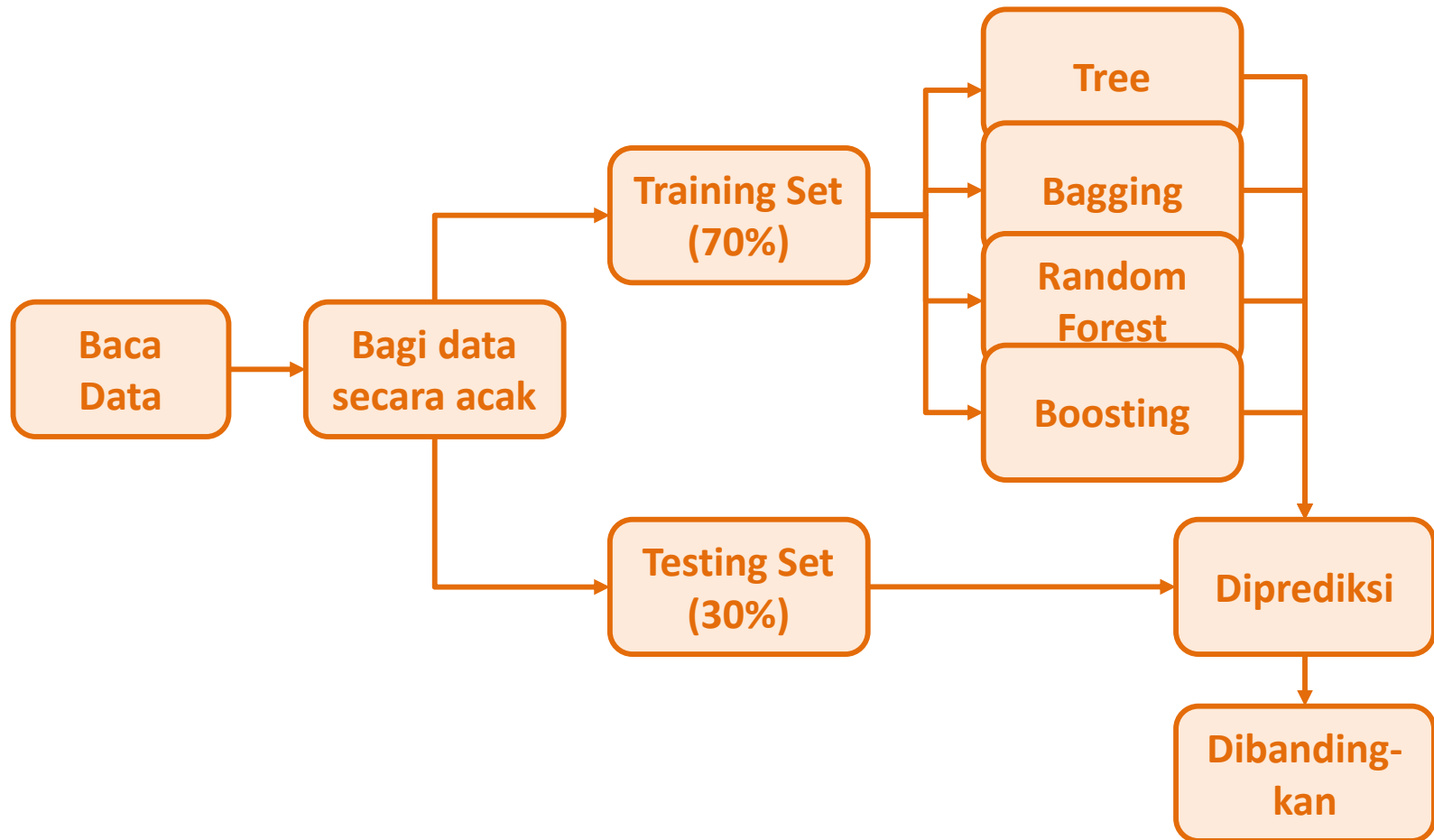
5. Normalize $D(x_i)$ so that $\sum_{i=1}^N D(x_i) = 1$

$$f_{final}(x) = \text{sign} [\sum \alpha_t h_t(x)]$$

Mari kita coba di R

- Bandingkan performa tree (pohon tunggal) dengan bagging, random forest, dan boosting....

Alur Analisis



```
#menjalankan algoritma boosting dan menilai kinerjanya
library(ada)
model.boost <- ada(Class~.,data=train,type="discrete")

pred.boost <- predict(model.boost,test,type="vector")

kinerja.boosting <- confusionMatrix(pred.boost,
test$Class, positive = "good")

kinerja.boosting
```



```
> kinerja.boosting
```

```
Confusion Matrix and Statistics
```

	Reference	
Prediction	bad	good
bad	29	0
good	8	67

```
Accuracy : 0.9231
```

```
95% CI : (0.854, 0.9662)
```

```
No Information Rate : 0.6442
```

```
P-Value [Acc > NIR] : 3.604e-11
```

```
Kappa : 0.8237
```

```
McNemar's Test P-Value : 0.01333
```

```
Sensitivity : 1.0000
```

```
Specificity : 0.7838
```

```
Pos Pred Value : 0.8933
```

```
Neg Pred Value : 1.0000
```

```
Prevalence : 0.6442
```

```
Detection Rate : 0.6442
```

```
Detection Prevalence : 0.7212
```

```
Balanced Accuracy : 0.8919
```

```
'Positive' Class : good
```


Apa itu Data dengan Kelas Tak Seimbang?

- Data dengan kelas tidak seimbang merujuk pada situasi dimana keberadaan amatan dari masing-masing kelas timpang jumlahnya.
- Sebagai contoh, kita barangkali memiliki 1000 buah amatan dimana kelas pertama sebanyak 800 amatan dan kelas kedua sebanyak 200 amatan, atau dengan rasio 4:1. Situasi lain dapat saja terjadi dengan ketimpangan yang jauh lebih tinggi.

Ketakseimbangan adalah masalah yang umum ditemui

- Data dengan kelas yang tidak seimbang jumlahnya merupakan masalah yang umum dijumpai.
 - Kasus kredit macet... non-performing loan hanya sekitar 2%-3%
 - Penawaran produk melalui telepon... yang merespon positif tidak lebih dari 1%
 - Kejadian terjangkitnya penyakit tertentu di masyarakat... sangat kecil proporsinya
- Kelas yang memiliki proporsi yang sedikit disebut sebagai kelas “minoritas”, sedangkan kelas yang proporsinya dominan disebut kelas “mayoritas”.

Accuracy Paradox

- Bayangkan kita punya data dimana perbandingan banyaknya amatan antara kelas 0 dan kelas 1 adalah 95:5
- Jika kita memperoleh model, dan dugaan dari model tersebut menghasilkan prediksi kelas 0 untuk semua amatan.
- Akurasinya 95%....
- Tapi model itu gagal memprediksi dengan benar satupun amatan dari kelas minoritas.

A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches

Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, *Member, IEEE*,
and Francisco Herrera, *Member, IEEE*

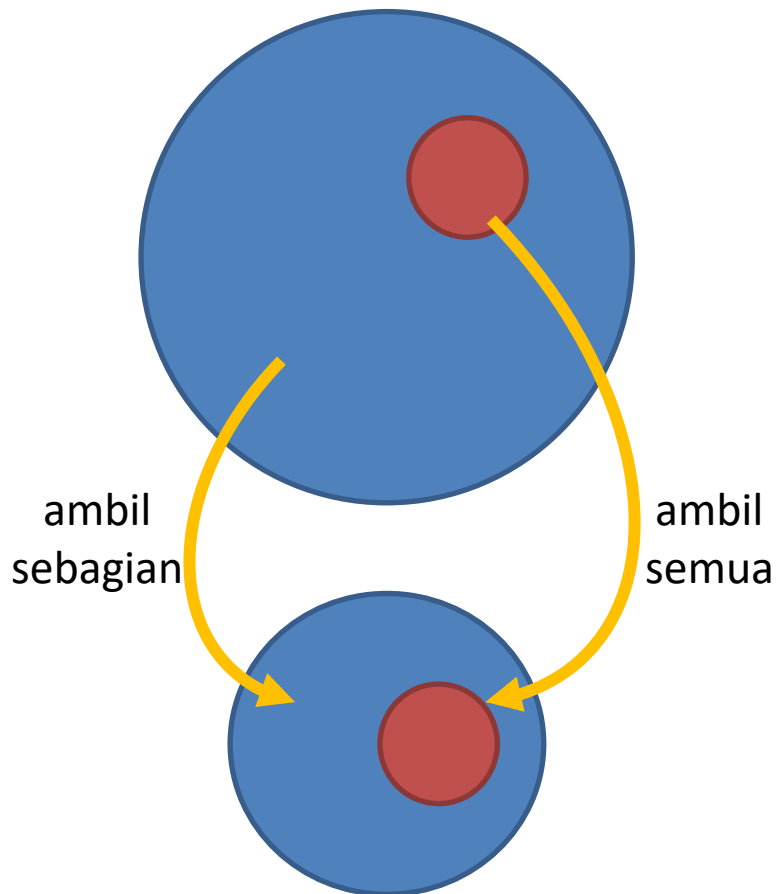
M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463-484, July 2012.

Beberapa pilihan

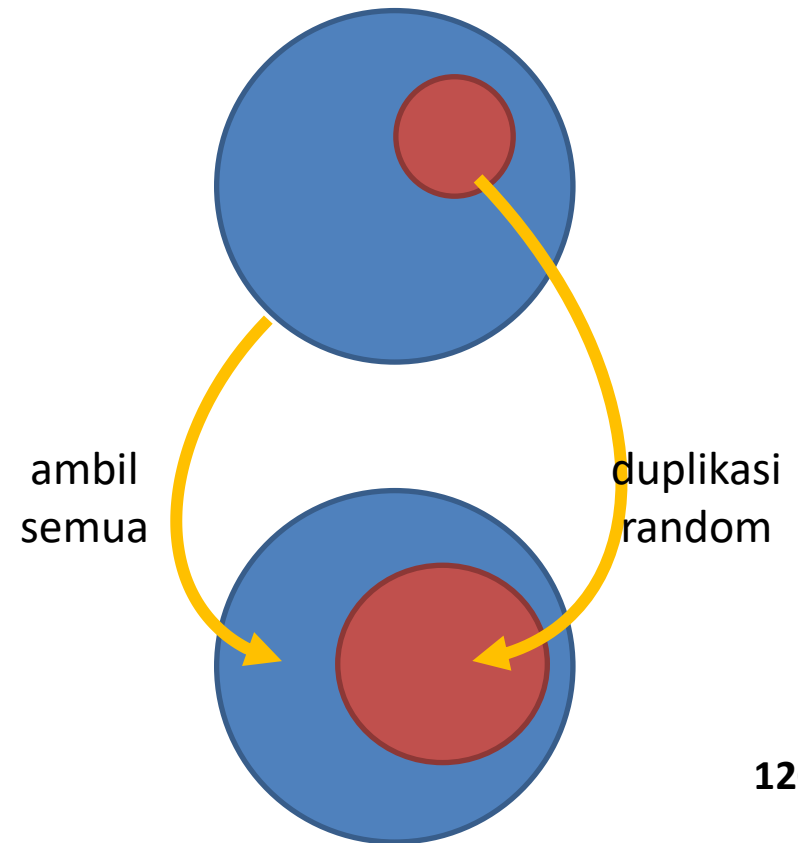
- Over-Bagging, gabungan antara oversampling dan Bagging
- Under-Sampling, gabungan antara undersampling dan Bagging
- EasyEnsemble , kombinasi undersampling dan Boosting
- RUSBoost, kombinasi undersampling dan Boosting
- dll

Undersampling dan Oversampling

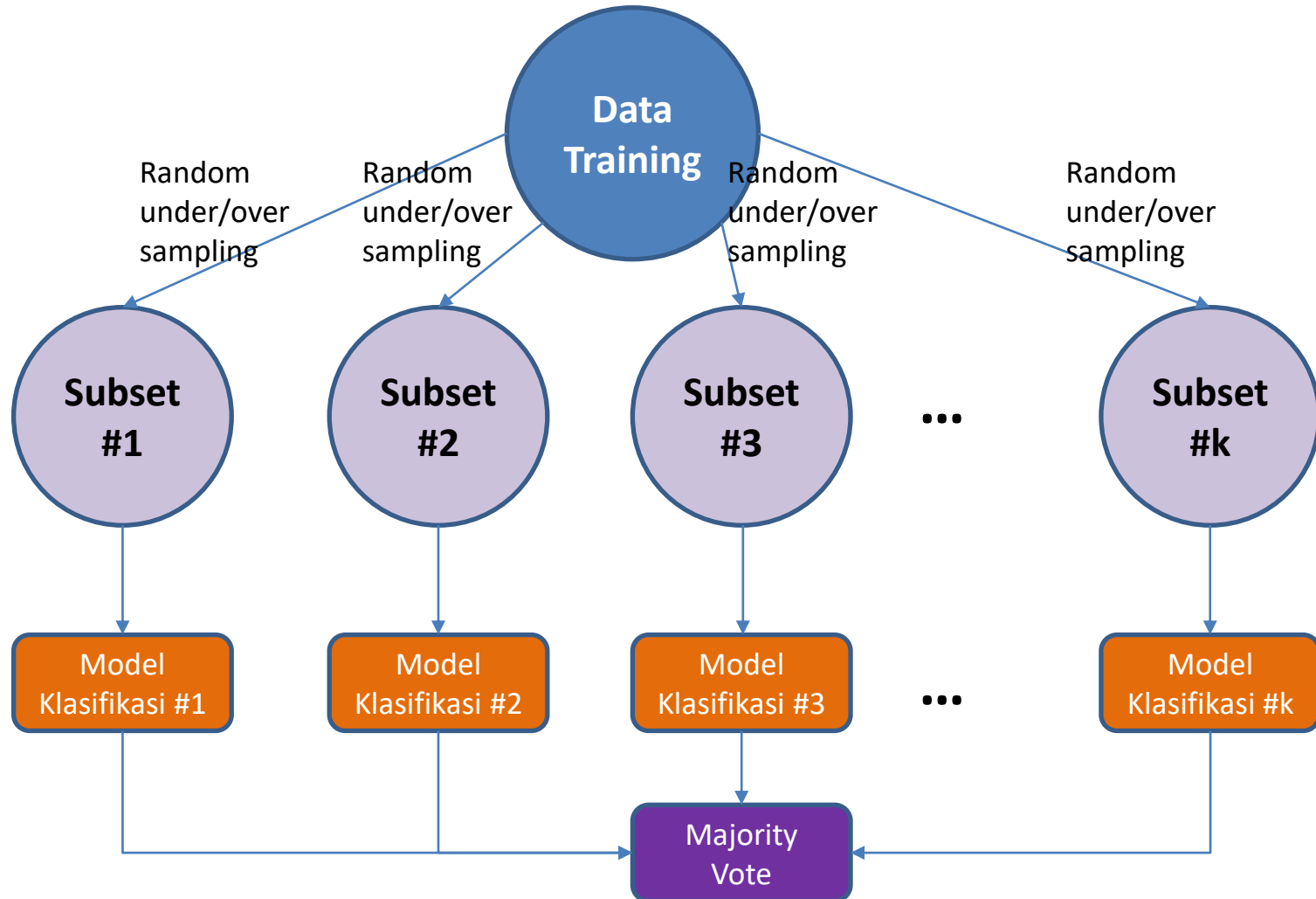
undersampling



oversampling



Under/Over-Bagging



EasyEnsemble

Algorithm 1 The EasyEnsemble algorithm.

- 1: {Input: A set of minority class examples \mathcal{P} , a set of majority class examples \mathcal{N} , $|\mathcal{P}| < |\mathcal{N}|$, the number of subsets T to sample from \mathcal{N} , and s_i , the number of iterations to train an AdaBoost ensemble H_i }
 - 2: $i \leftarrow 0$
 - 3: **repeat**
 - 4: $i \leftarrow i + 1$
 - 5: Randomly sample a subset \mathcal{N}_i from \mathcal{N} , $|\mathcal{N}_i| = |\mathcal{P}|$.
 - 6: Learn H_i using \mathcal{P} and \mathcal{N}_i . H_i is an AdaBoost ensemble with s_i weak classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$. The ensemble's threshold is θ_i , i.e.
$$H_i(x) = \text{sgn} \left(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \theta_i \right).$$
 - 7: **until** $i = T$
 - 8: Output: An ensemble:
$$H(x) = \text{sgn} \left(\sum_{i=1}^T \sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \sum_{i=1}^T \theta_i \right).$$
-

RUSBoost

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). **RUSBoost: A hybrid approach to alleviating class imbalance**. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1), 185-197.

Algorithm RUSBoost

Given: Set S of examples $(x_1, y_1), \dots, (x_m, y_m)$ with minority class $y^r \in Y$, $|Y| = 2$

Weak learner, *WeakLearn*

Number of iterations, T

Desired percentage of total instances to be represented by the minority class, N

1 Initialize $D_1(i) = \frac{1}{m}$ for all i .

2 Do for $t = 1, 2, \dots, T$

a Create temporary training dataset S'_t with distribution D'_t using random undersampling

b Call *WeakLearn*, providing it with examples S'_t and their weights D'_t .

c Get back a hypothesis $h_t : X \times Y \rightarrow [0, 1]$.

d Calculate the pseudo-loss (for S and D_t):

$$\epsilon_t = \sum_{(i,y):y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y)).$$

e Calculate the weight update parameter:

$$\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}.$$

f Update D_t :

$$D_{t+1}(i) = D_t(i)\alpha_t^{\frac{1}{2}(1+h_t(x_i, y_i)-h_t(x_i, y:y \neq y_i))}.$$

g Normalize D_{t+1} : Let $Z_t = \sum_i D_{t+1}(i)$.

$$D_{t+1}(i) = \frac{D_{t+1}(i)}{Z_t}.$$

3 Output the final hypothesis:

$$H(x) = \operatorname{argmax}_{y \in Y} \sum_{t=1}^T h_t(x, y) \log \frac{1}{\alpha_t}.$$

Terima Kasih