Ensemble Learning

disusun oleh:

Bagus Sartono

bagusco@gmail.com

0852-1523-1823

Program Studi Statistika

Jurusan Matematika - FMIPA

Universitas Tadulako

Departemen Statistika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Pertanian Bogor

2018

Bagus Sartono

statistician and data scientist

Pengalaman Kerja

2000 – sekarang	Dosen, Departemen Statistika IPB
2012 – sekarang	Dosen, Sekolah Bisnis IPB
2015 – sekarang	Data Science Advisor, Starcore
2014	Technical Advisor, MarkPlus Insight
2014	Consultant, CIFOR
2000 – sekarang	Tenaga Ahli/Trainer,/Narasumber di OJK, Bank Indonesia, Kementerian Keuangan, Bank Mandiri, LIPI, InfoKomputer, SAS Indonesia, Ganesha Cipta Informatika

Pendidikan

2000	Sarjana Sains, Statistika IPB
2004	Magister Sains, Statistika IPB
2012	PhD in Applied Economics, Universiteit Antwerpen

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* yang memerlukan *short-list* calon pelanggan untuk dihubungi dan ditawari produk. *Short-list* tersebut umumnya diperoleh dari *list* yang sangat panjang dan memuat 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 *campaign* via SMS (*short message service*) oleh berbagai perusahaan retail.

Naskah:
Bagus Sartono
Dosen di Departemen Statistika FMIPA IPB
bagussr@gmail.com, bagussr@ipb.ac.id
Ilustrasi Foto:
Thinkstock.com

Tersedia 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 penentuan 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*.

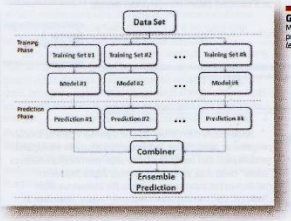
Kriteria yang kedua adalah terkait dengan kesederhanaan model. Secara naluriah, model yang disukai adalah model yang lebih ringkas, menggunakan *predictor* yang lebih sedikit, atau bentuk-bentuk fungsi yang lebih sederhana. Kriteria kedua ini dikenal sebagai *complexity cost*. Ukuran yang tergolong dalam kriteria ini meliputi banyaknya parameter dalam model, banyaknya simpul pada *tree* dan *neural network*, serta derajat *polinomial* 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. Pasalnya, 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. Mekanisma dasar pendekatan *ensemble learning* (Tsiakis, 2015).

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*. Theodosios Tsiakis (2015) dalam buku yang diedit-nya 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 prediksi 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* menggabungkan 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 mengandalkan 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 tertuang pada berbagai paper ilmiah, namun juga dapat dilihat pada penyelesaian kasus-kasus aplikatif seperti yang dapat dilihat pada kompetisi *data science* Kaggle (https://www.kaggle.com/). Kompetisi ini terbuka bagi pegiat *data science* dan *data mining* untuk memberikan solusi prediktif dari kasus-kasus yang disampaikan oleh banyak perusahaan besar berskala internasional.

Setiap tim atau individu dipersilakan 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 berkembang 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

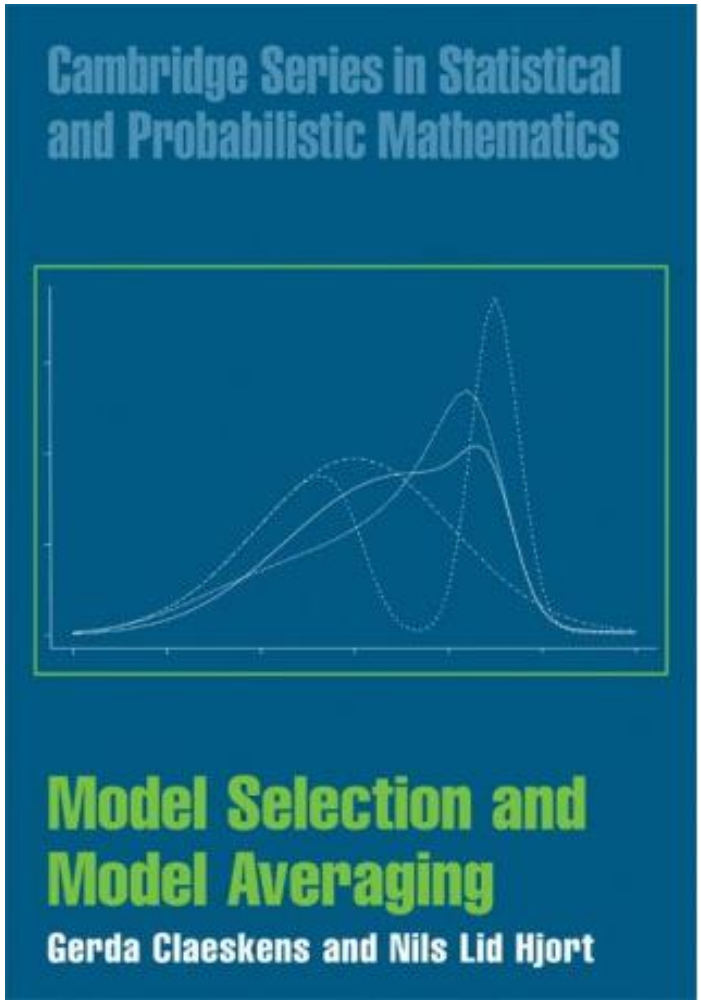
- Pengantar
- Classification Tree [optional]
- Bagging, Random Forest
- Boosting
- Lain-lain [optional]:
  - Ensemble untuk pemodelan klasifikasi pada data tidak seimbang
  - Ensemble of Ensembles

Prinsip Dasar

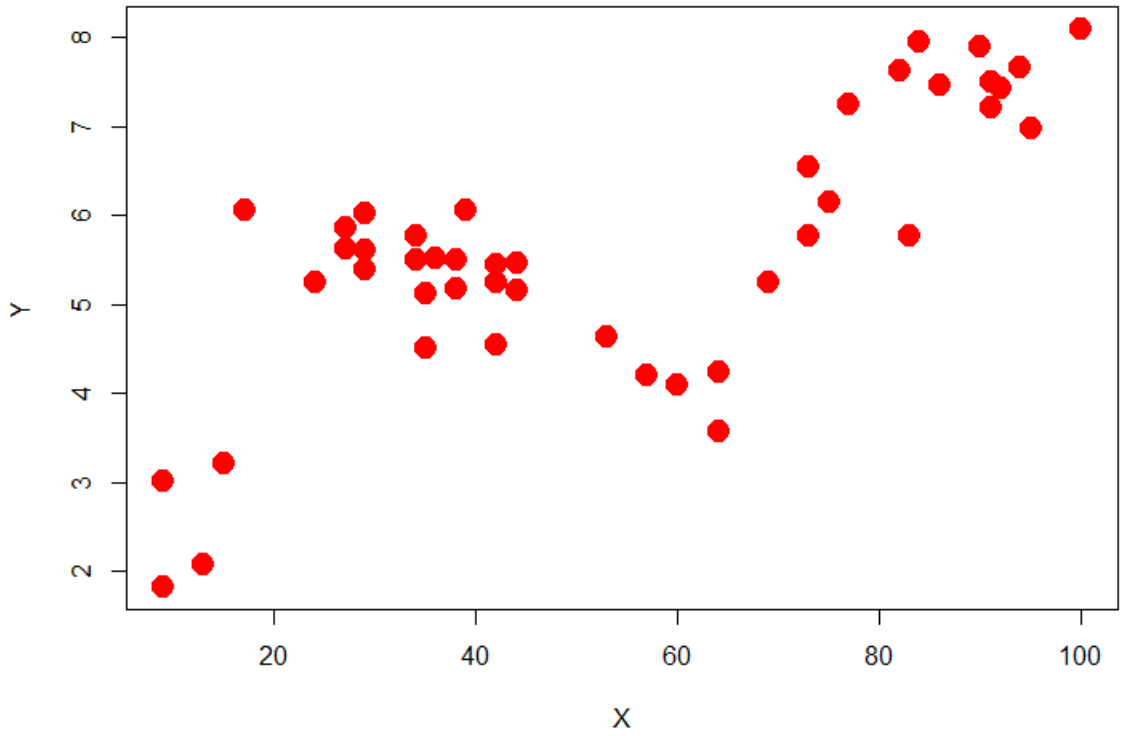
vs

single expert

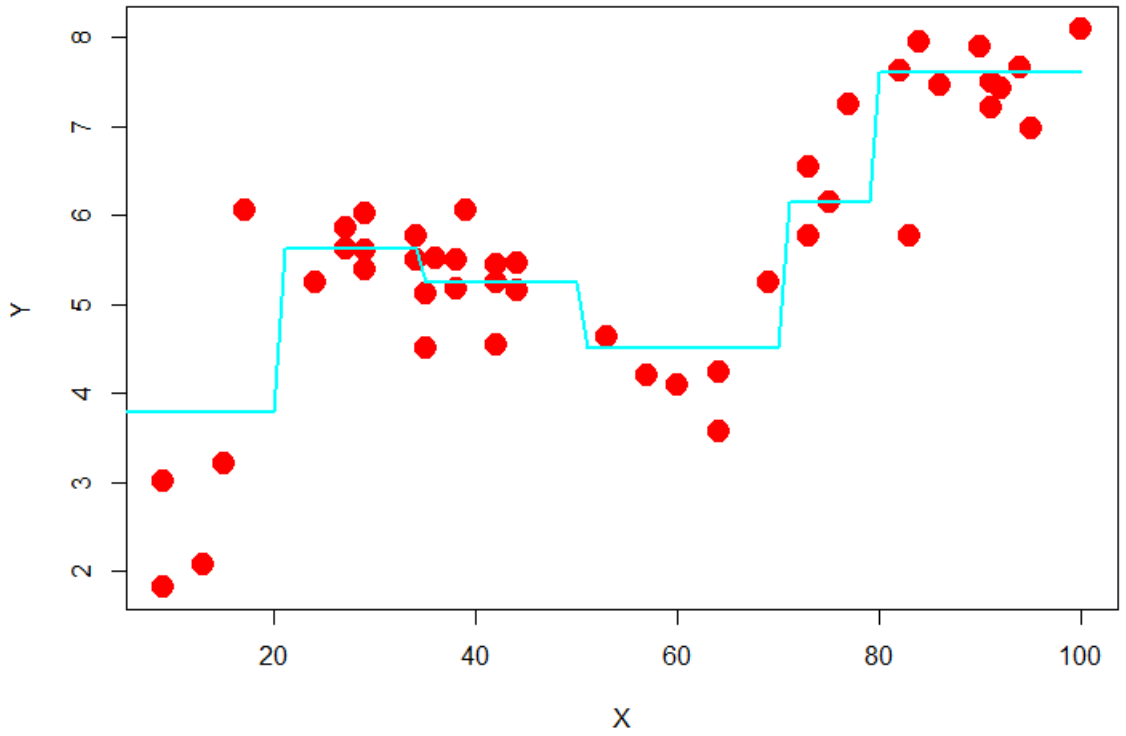
a team of experts


Cambridge Series in Statistical and Probabilistic Mathematics
Model Selection and Model Averaging
Gerda Claeskens and Nils Lid Hjort

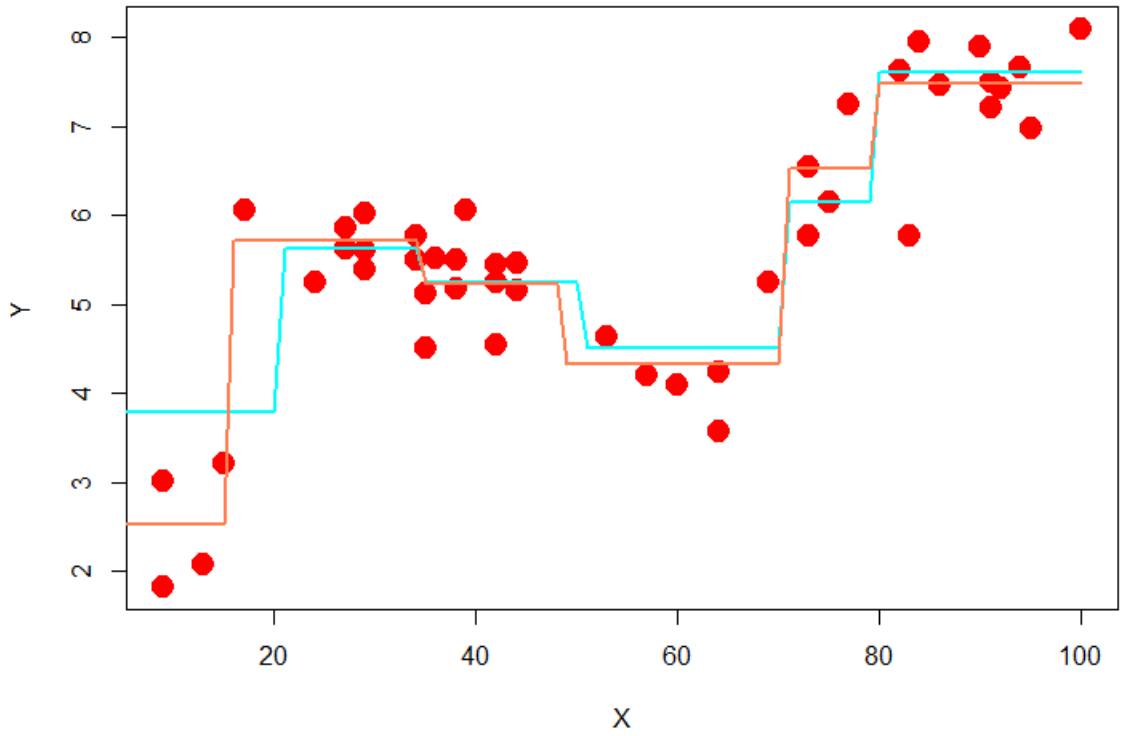
Motivating Example #1



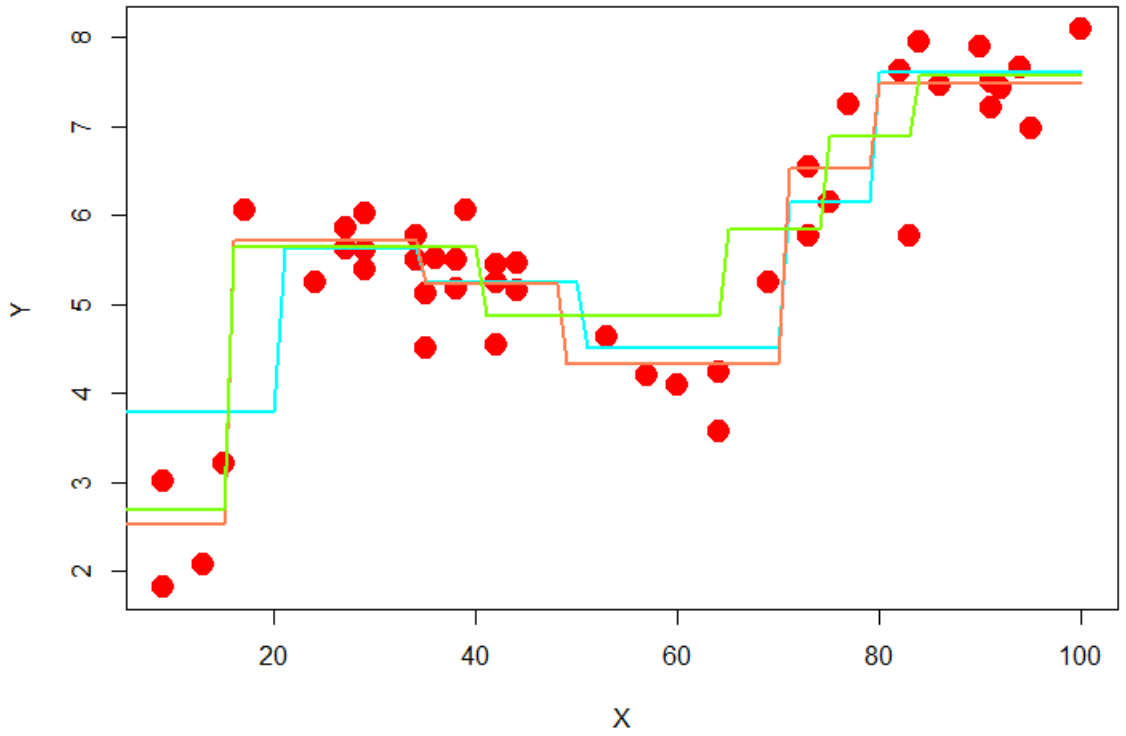
Motivating Example #1



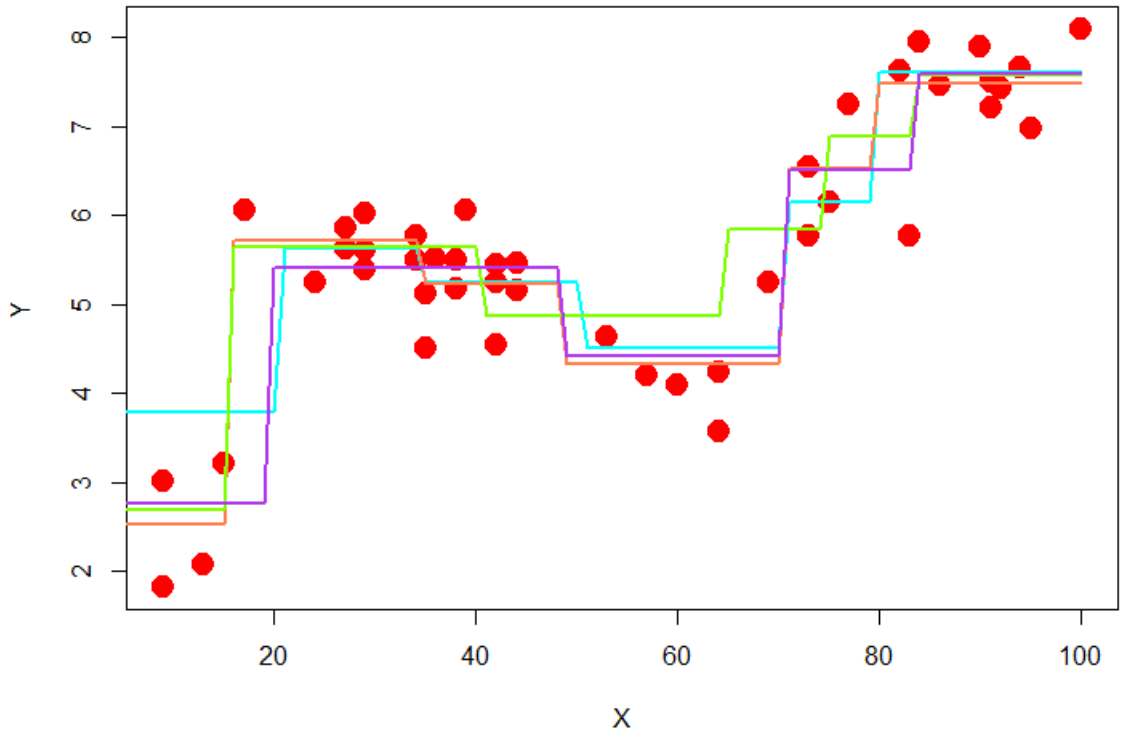
Motivating Example #1



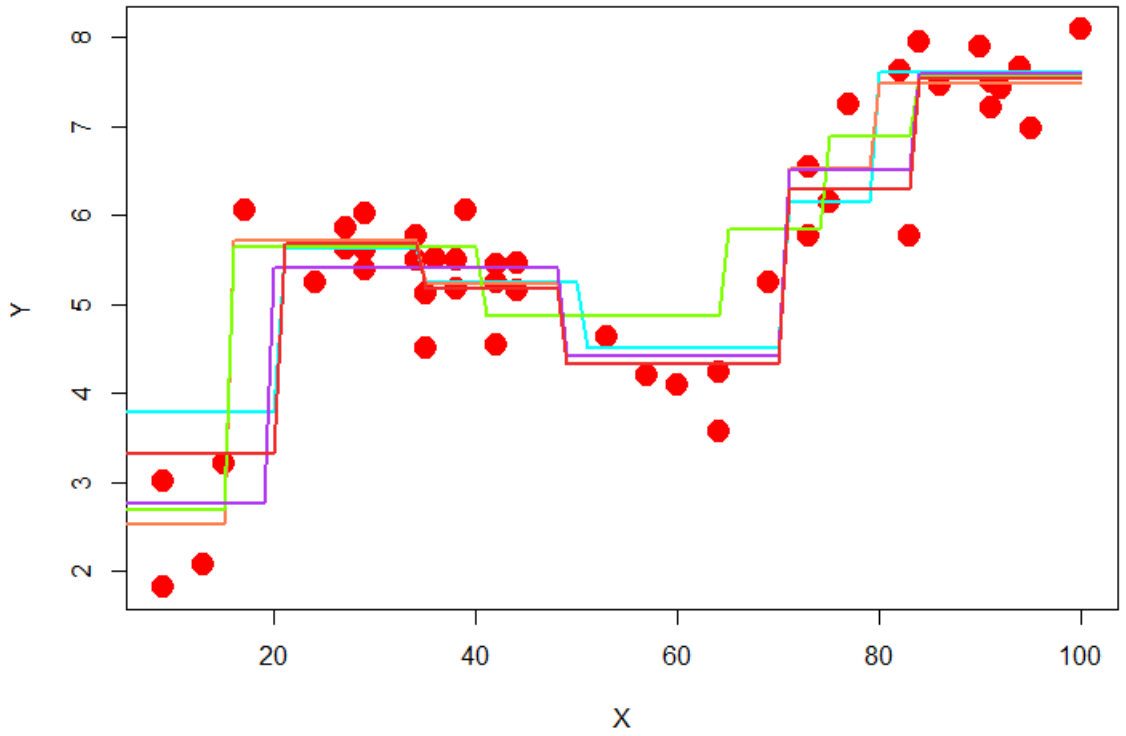
Motivating Example #1



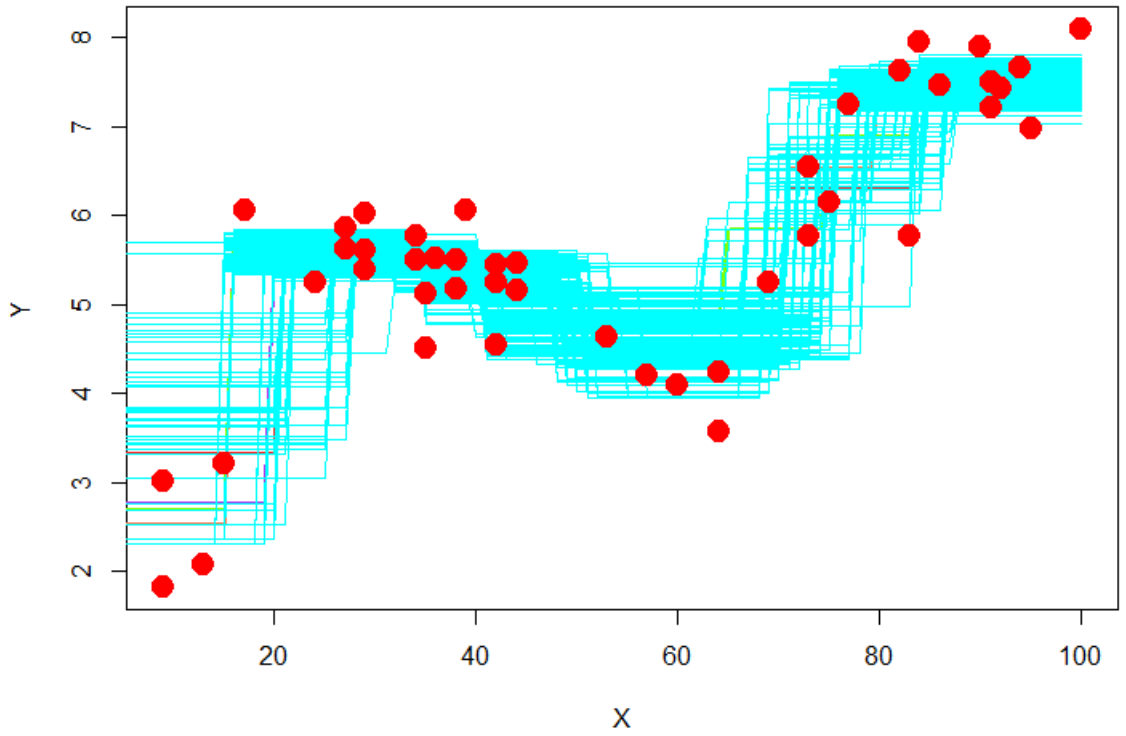
Motivating Example #1



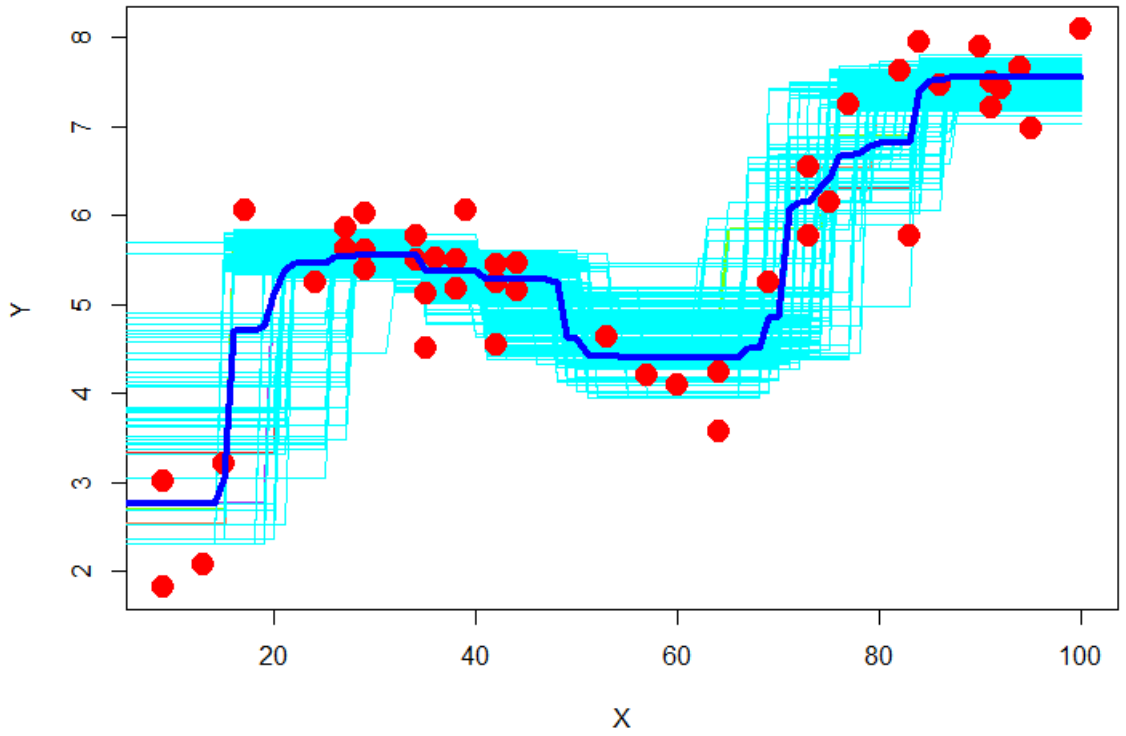
Motivating Example #1



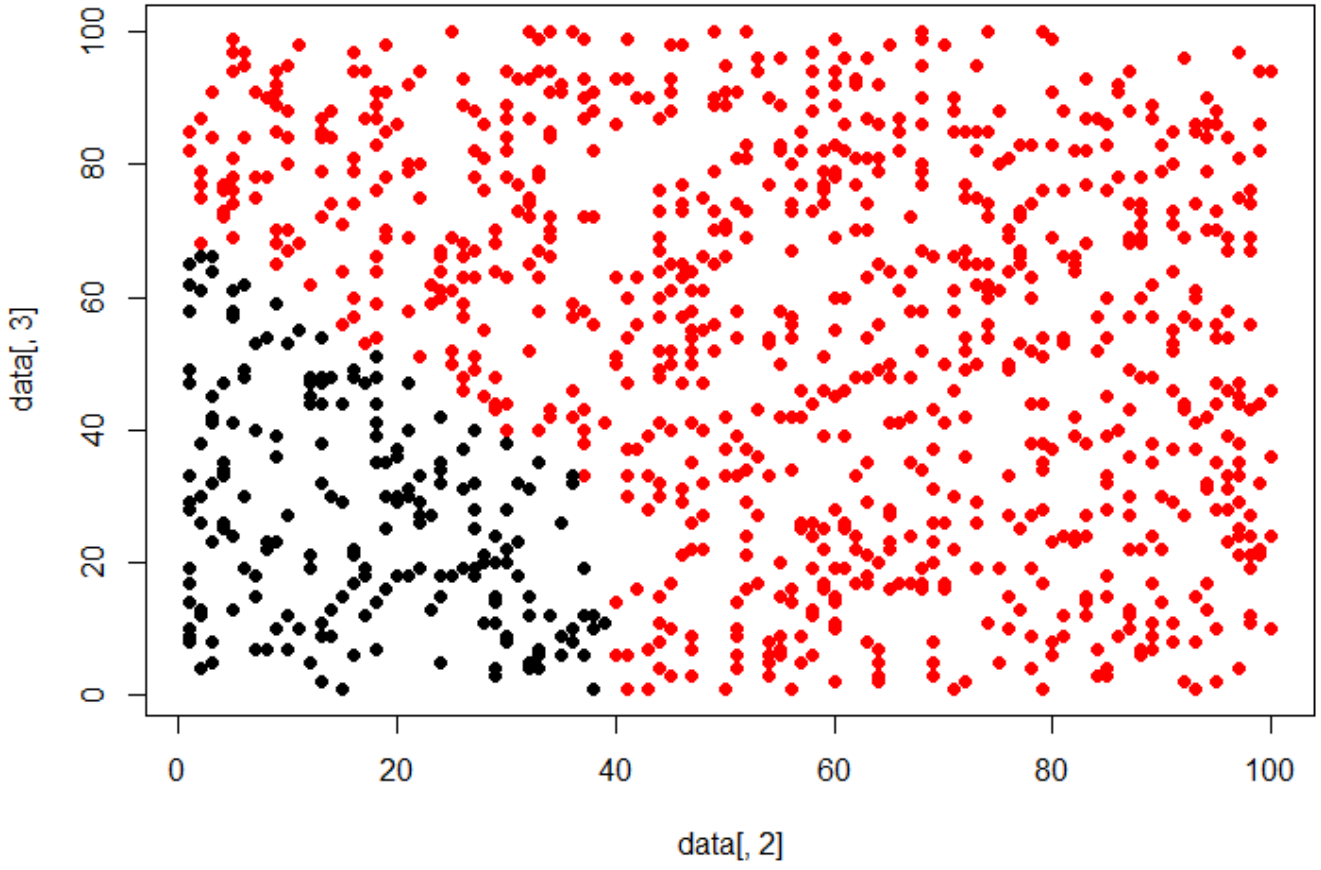
Motivating Example #1



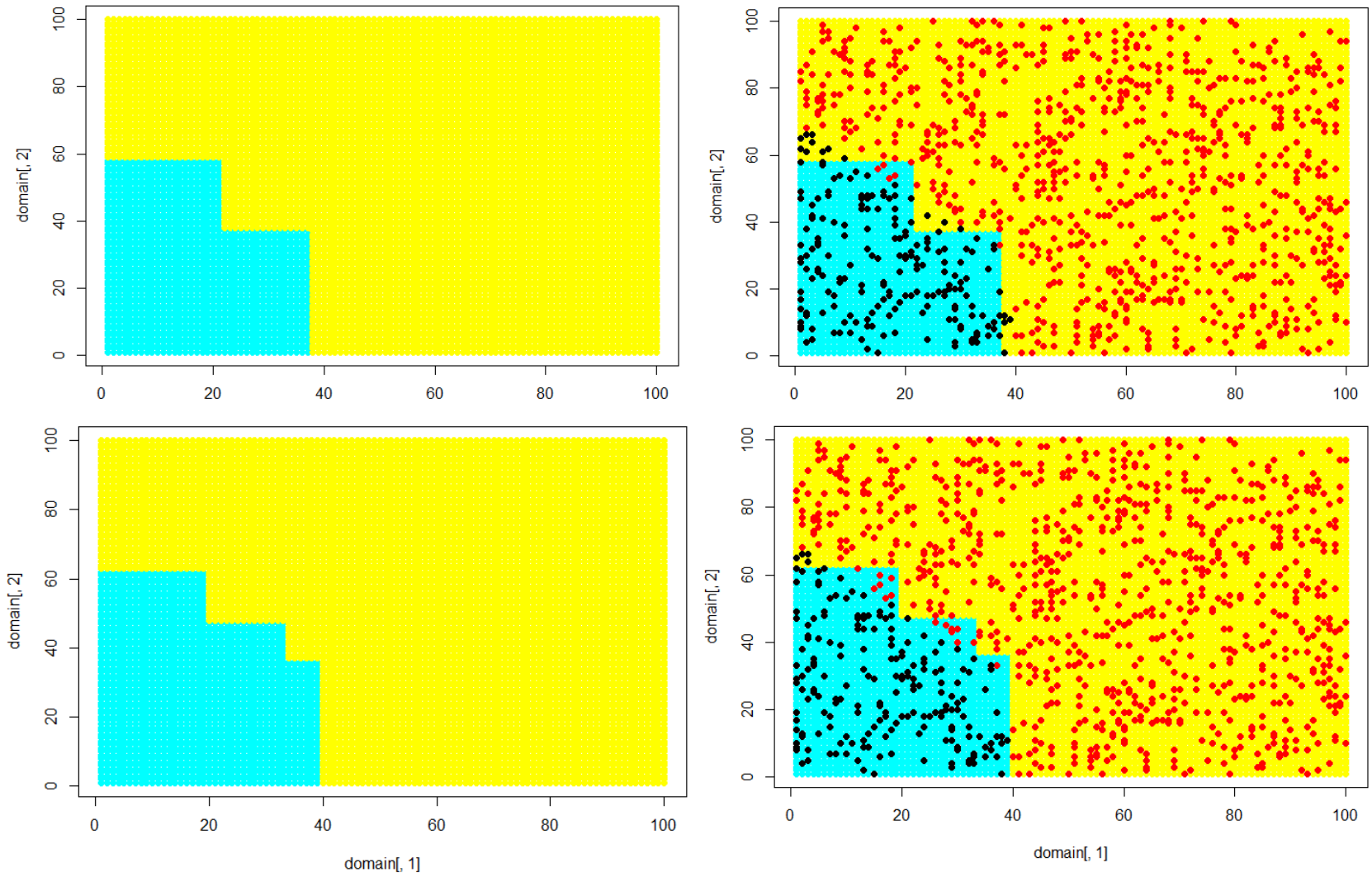
Motivating Example #1



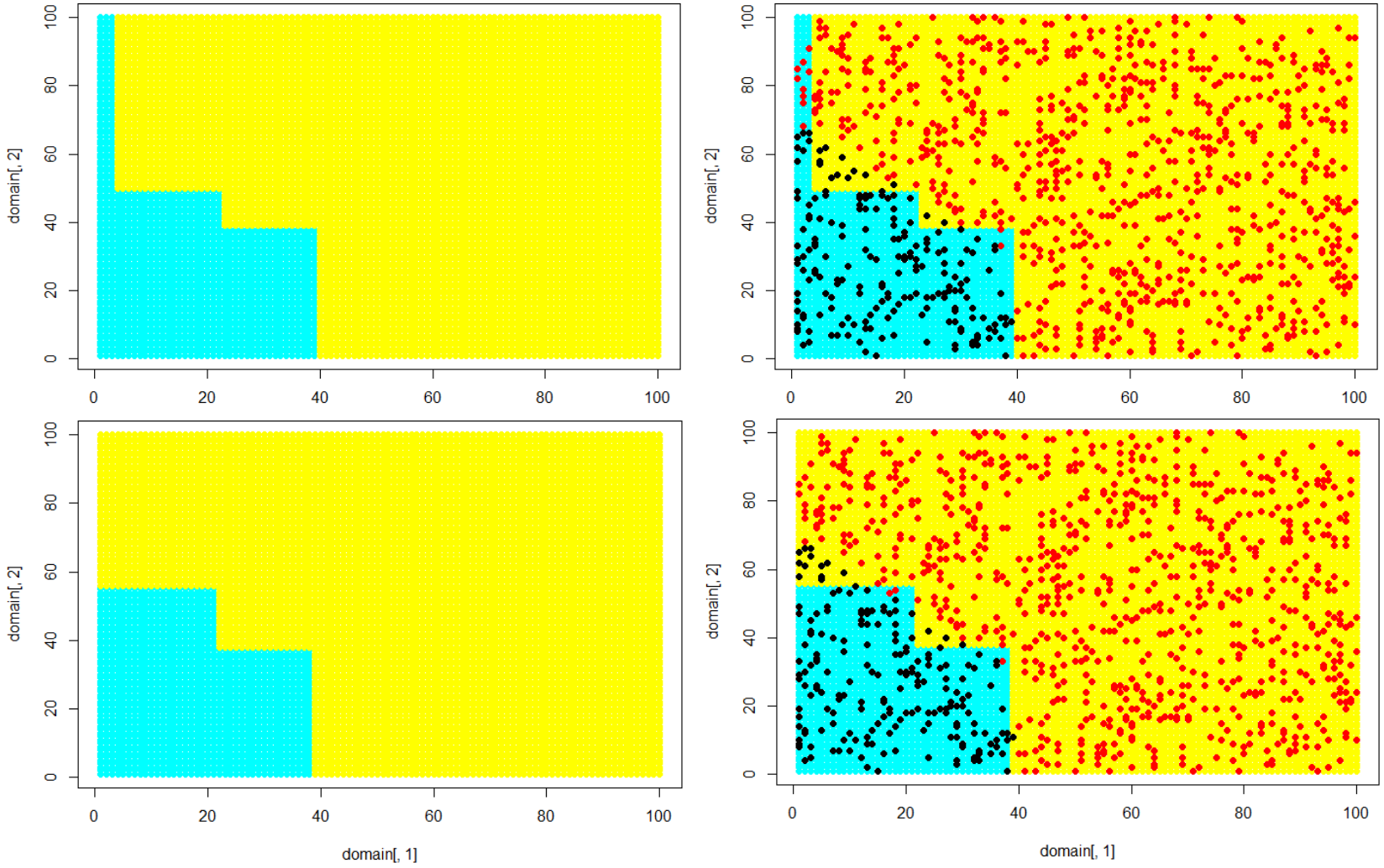
Motivating Example #2



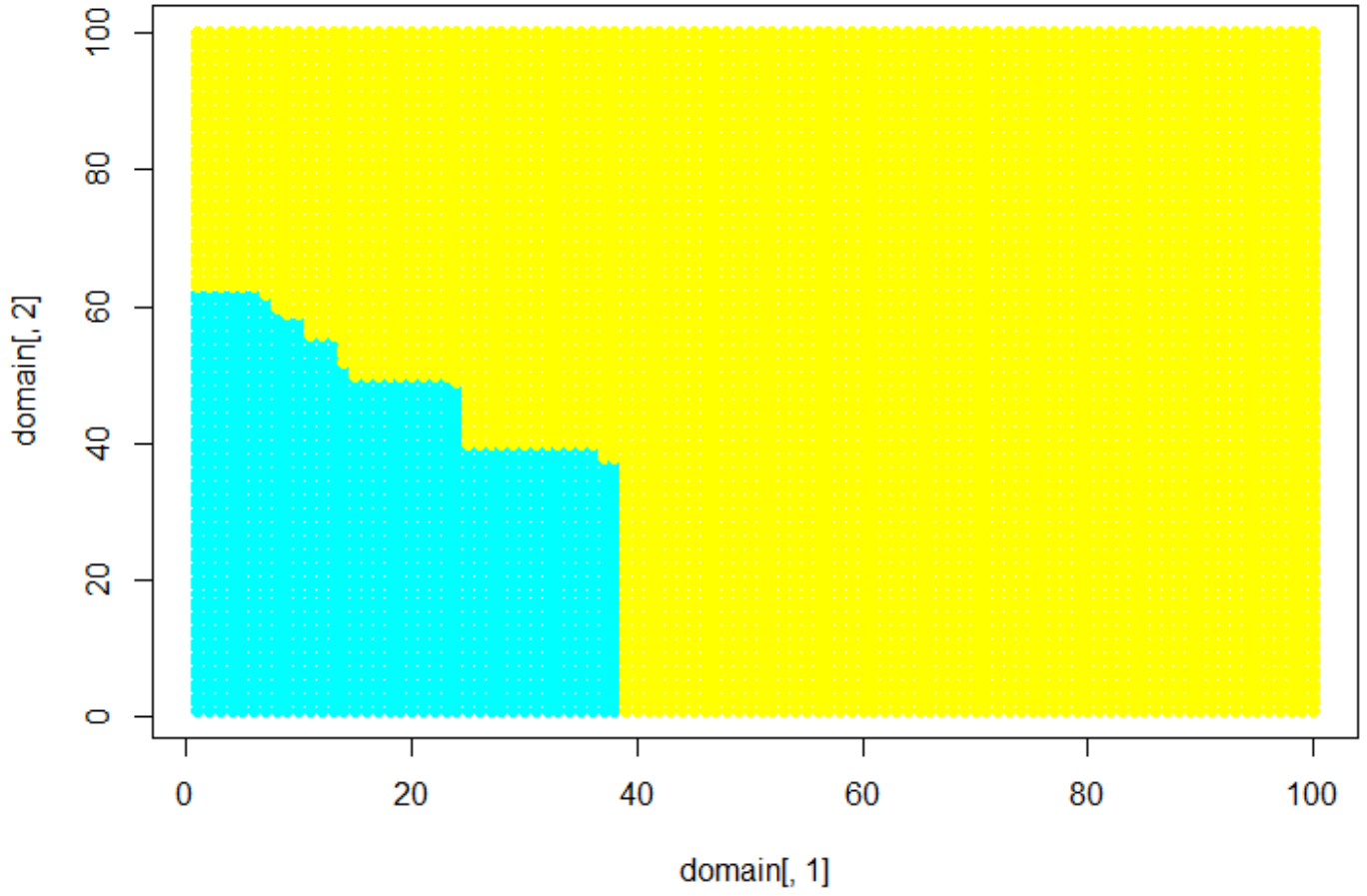
Motivating Example #2



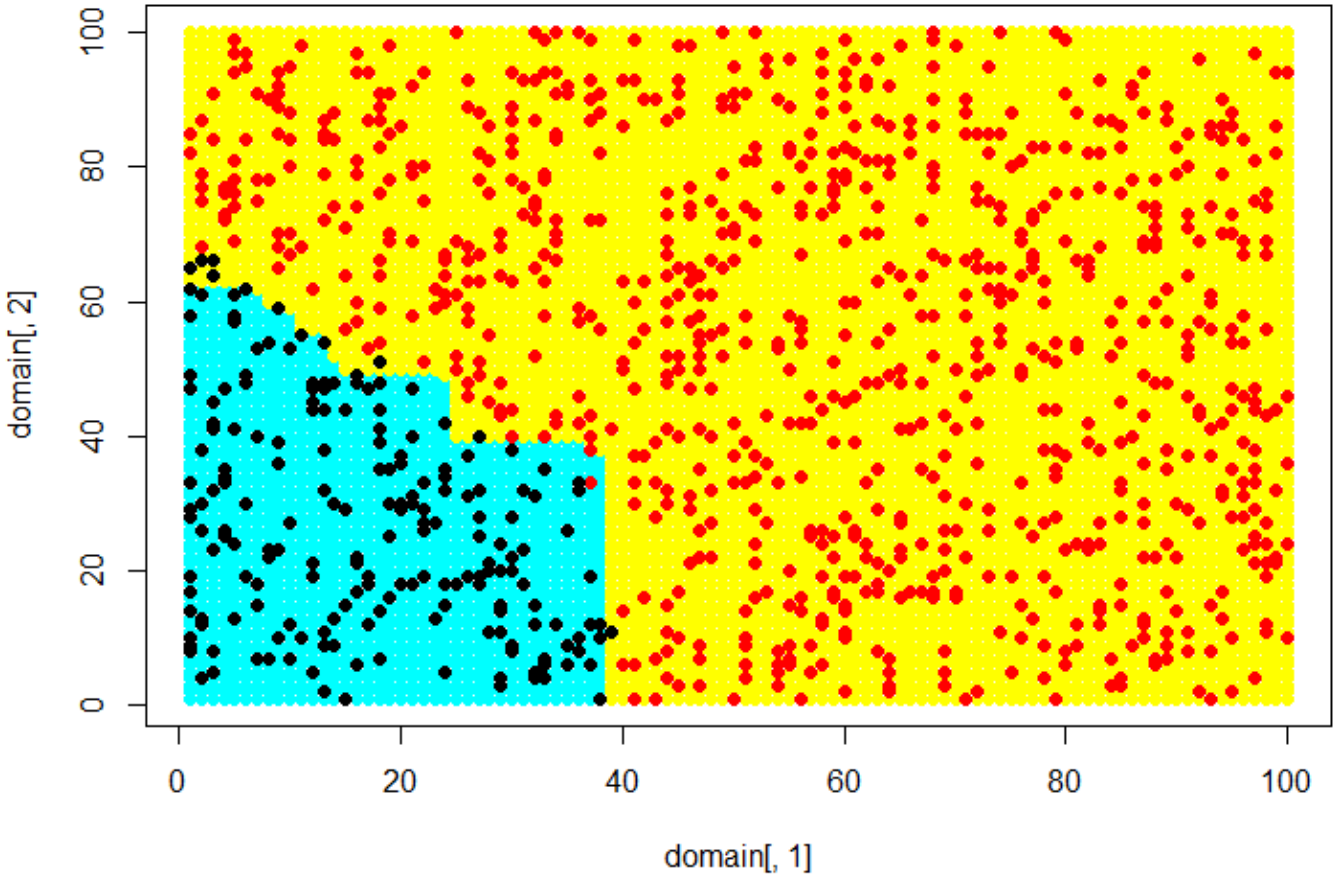
Motivating Example #2



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?

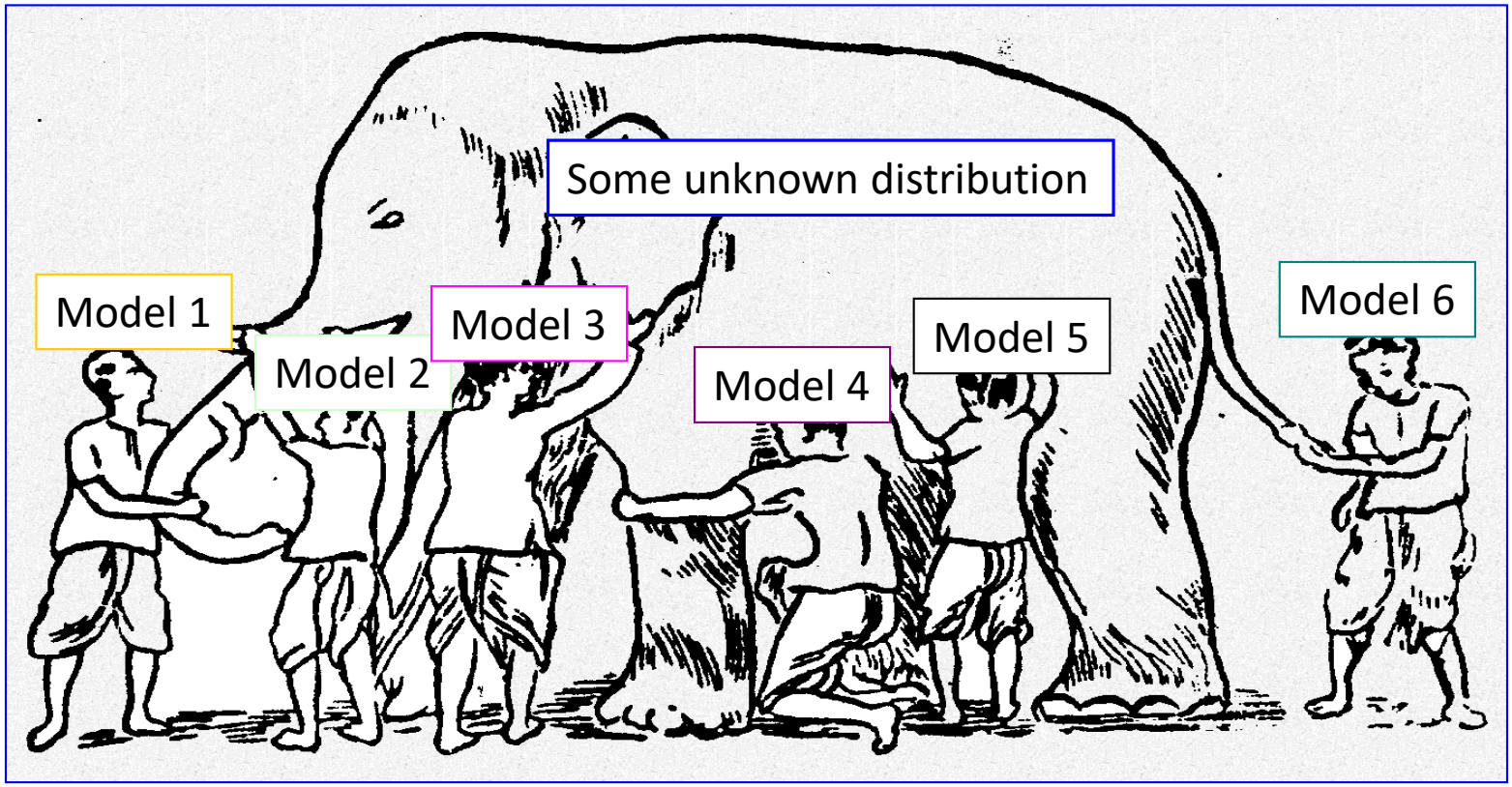
- Bisakah kita gabungkan kedua metode itu agar akurasinya meningkat?

→ 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?

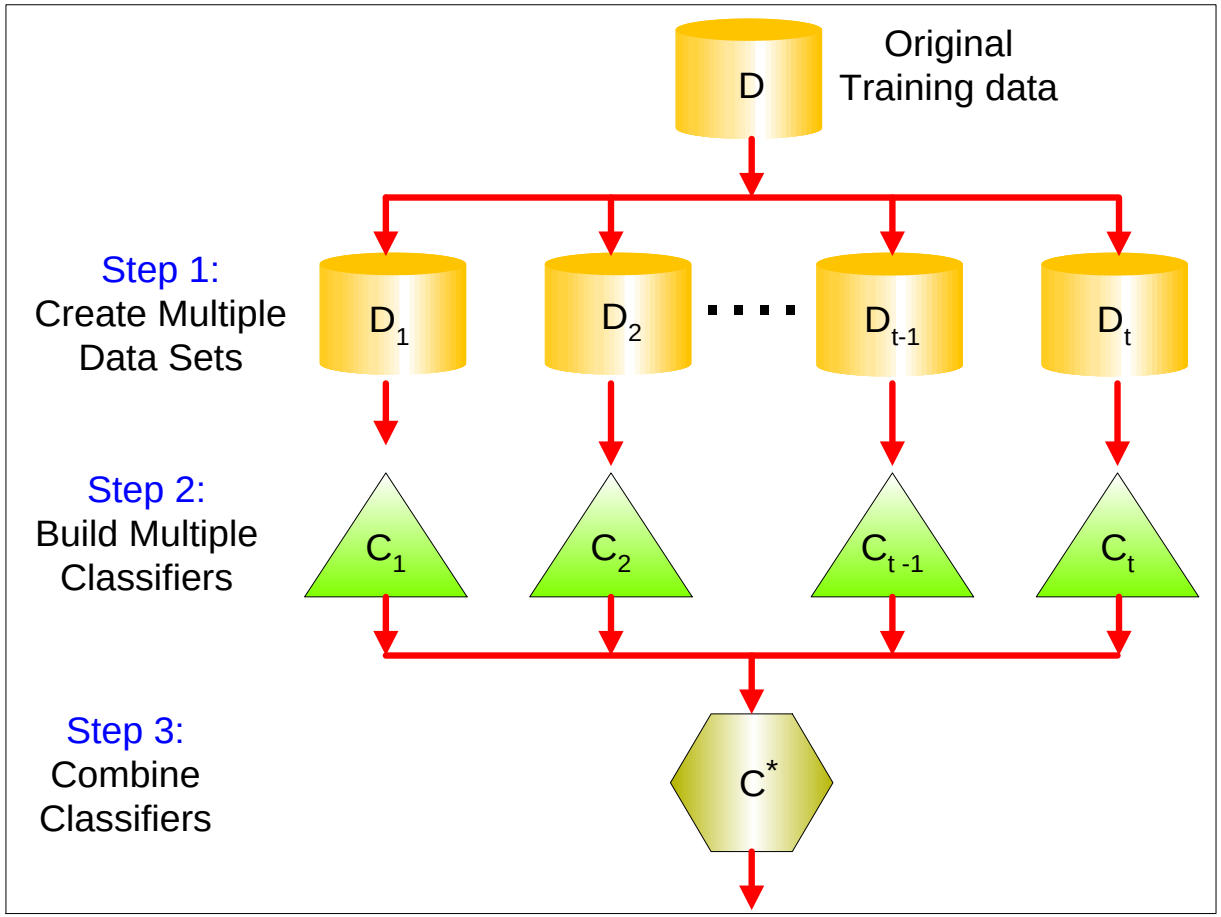
- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

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

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


D
Original
Training data
Step 1:
Create Multiple
Data Sets
D_1
D_2
D_{t-1}
D_t
Step 2:
Build Multiple
Classifiers
C_1
C_2
C_{t-1}
C_t
Step 3:
Combine
Classifiers
C^*

Pohon Klasifikasi (Classification Tree)

Apa itu Klasifikasi

Merriam-Webster

m-w.com

systematic arrangement in groups or categories according to established criteria

WIKIPEDIA
The Free Encyclopedia

identifying to which of a set
ub-populations) a new
ongs, on the basis of a training
aining observations (or
se category membership is

Persetujuan aplikasi pembiayaan (kredit)

CREDIT SCORING

Bank atau Lembaga Pembiayaan berkepentingan untuk menyeleksi calon nasabah pembiayaan (kredit).

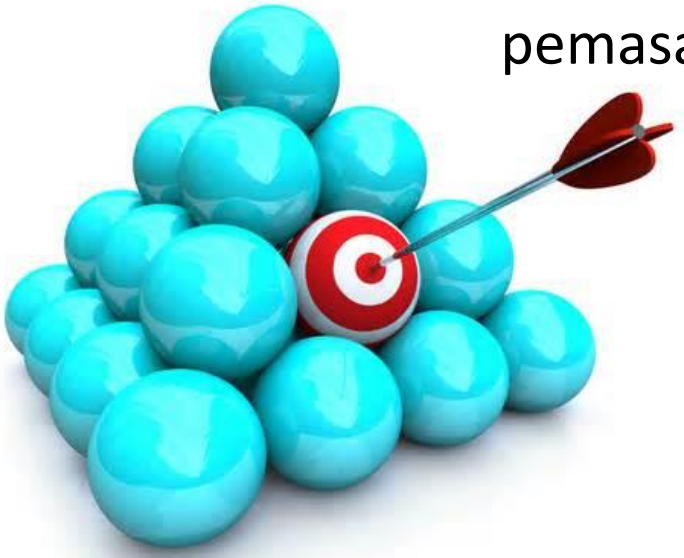
Berdasarkan data karateristik resiko (demografi, income, perilaku selama ini), calon nasabah diklasifikasikan menjadi high-risk atau low-risk.

Klasifikasi itu selanjutnya digunakan untuk memutuskan untuk **menerima** (to approve) atau **menolak** (to reject) aplikasi yang masuk.

Penentuan target pemasaran

Up-Sell, Cross-Sell, Direct Campaign

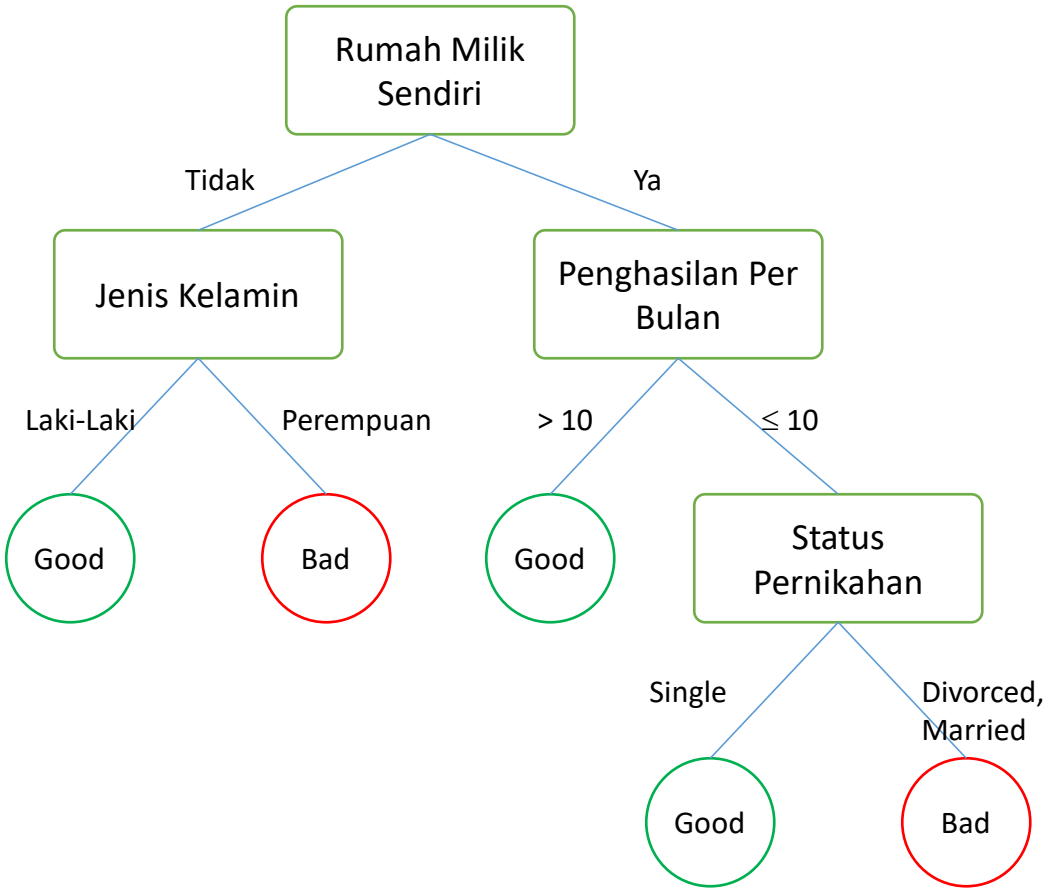
Beberapa perusahaan memiliki data base pelanggan yang bisa dijadikan target pemasaran produk tertentu.



Biaya yang besar dapat dihindari dengan hanya melakukan penawaran kepada pelanggan dengan potensi ketertarikan (prospek) yang besar.

Diperlukan proses klasifikasi terhadap database pelanggan untuk memisahkan pelanggan **potensial** dan yang **tidak potensial**.

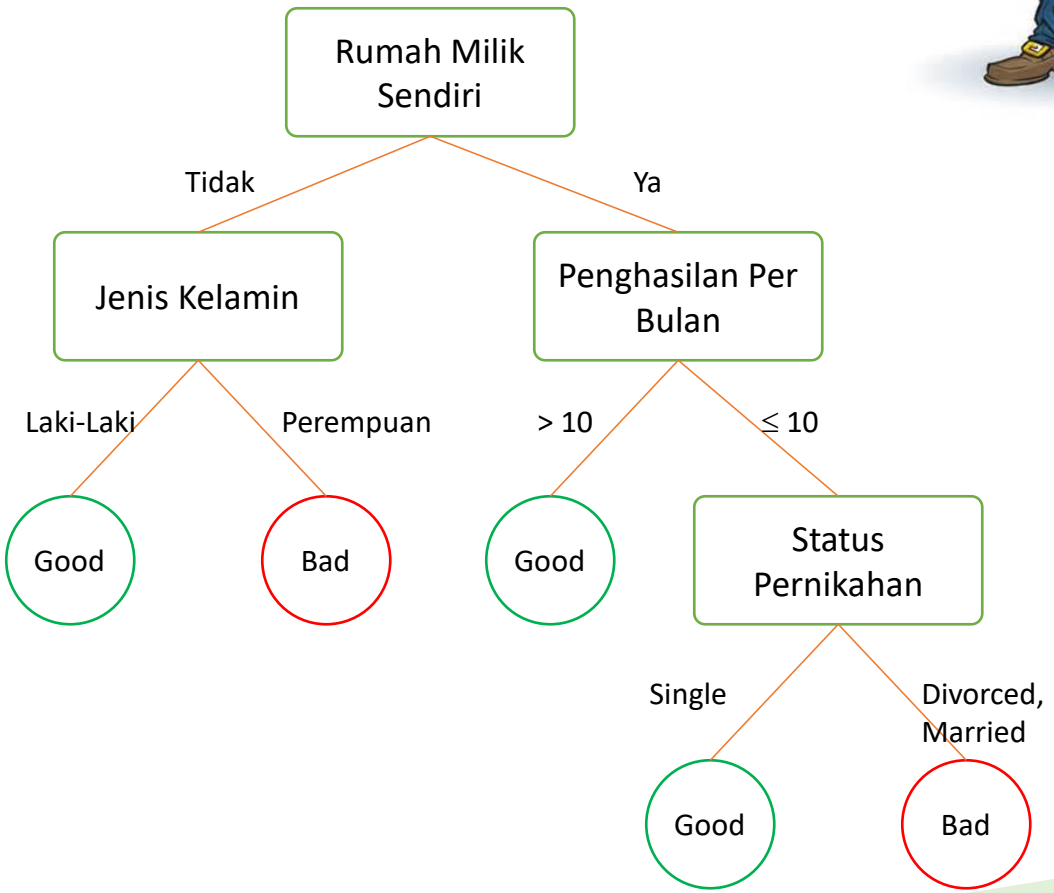
Ilustrasi penggunaan pohon klasifikasi



Ilustrasi penggunaan pohon klasifikasi

Profil:

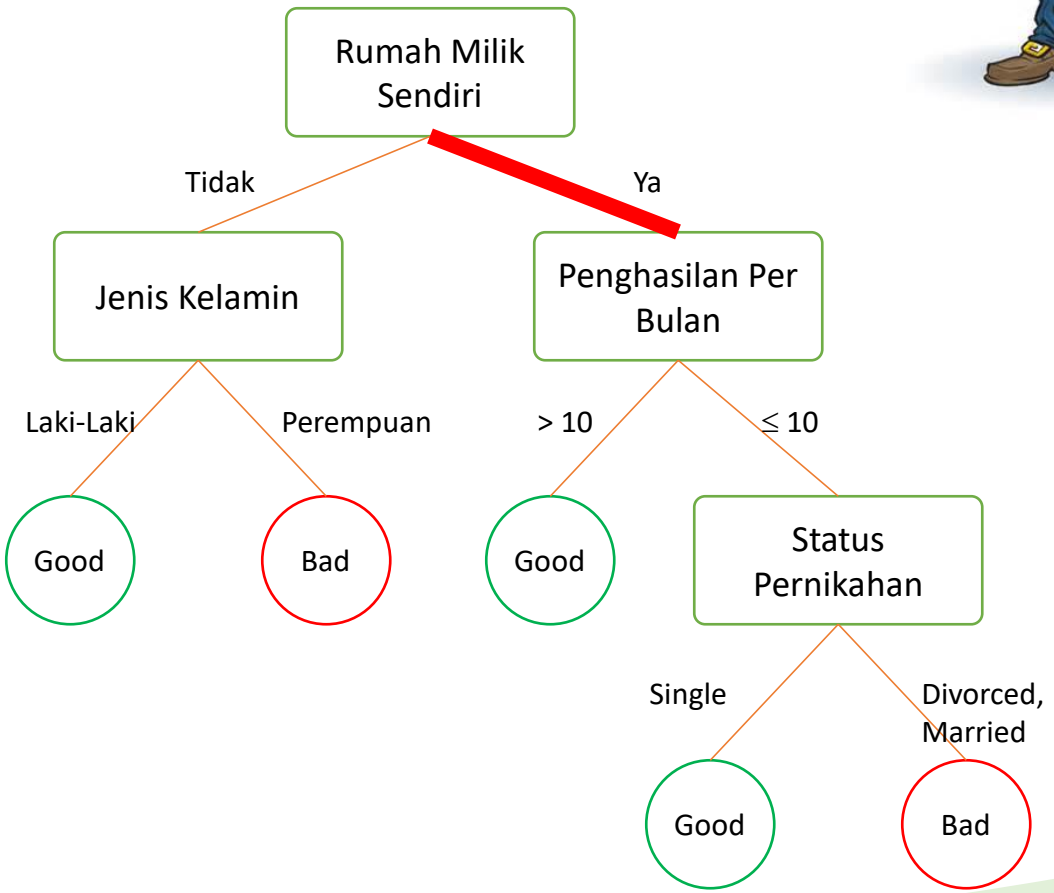
- Pria
- Rumah Sendiri
- Penghasilan 8 juta per bulan
- Bujangan



Ilustrasi penggunaan pohon klasifikasi

Profil:

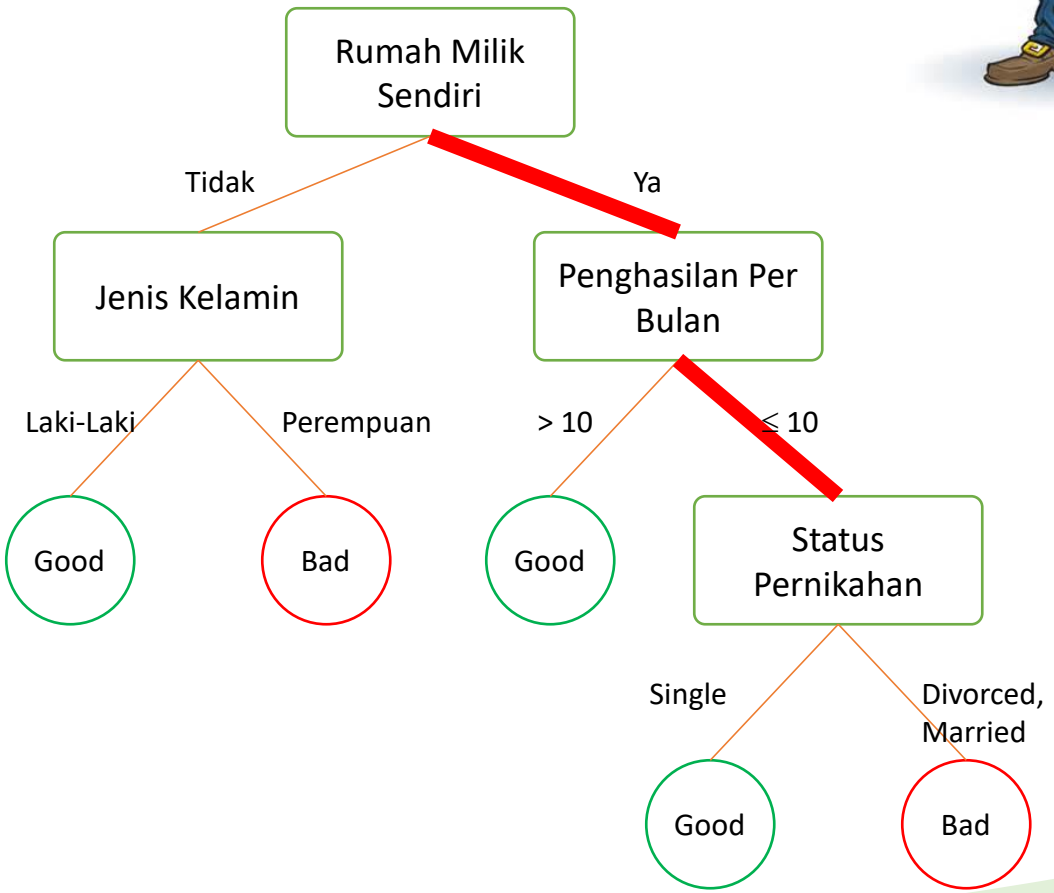
- Pria
- **Rumah Sendiri**
- Penghasilan 8 juta per bulan
- Bujangan



Ilustrasi penggunaan pohon klasifikasi

Profil:

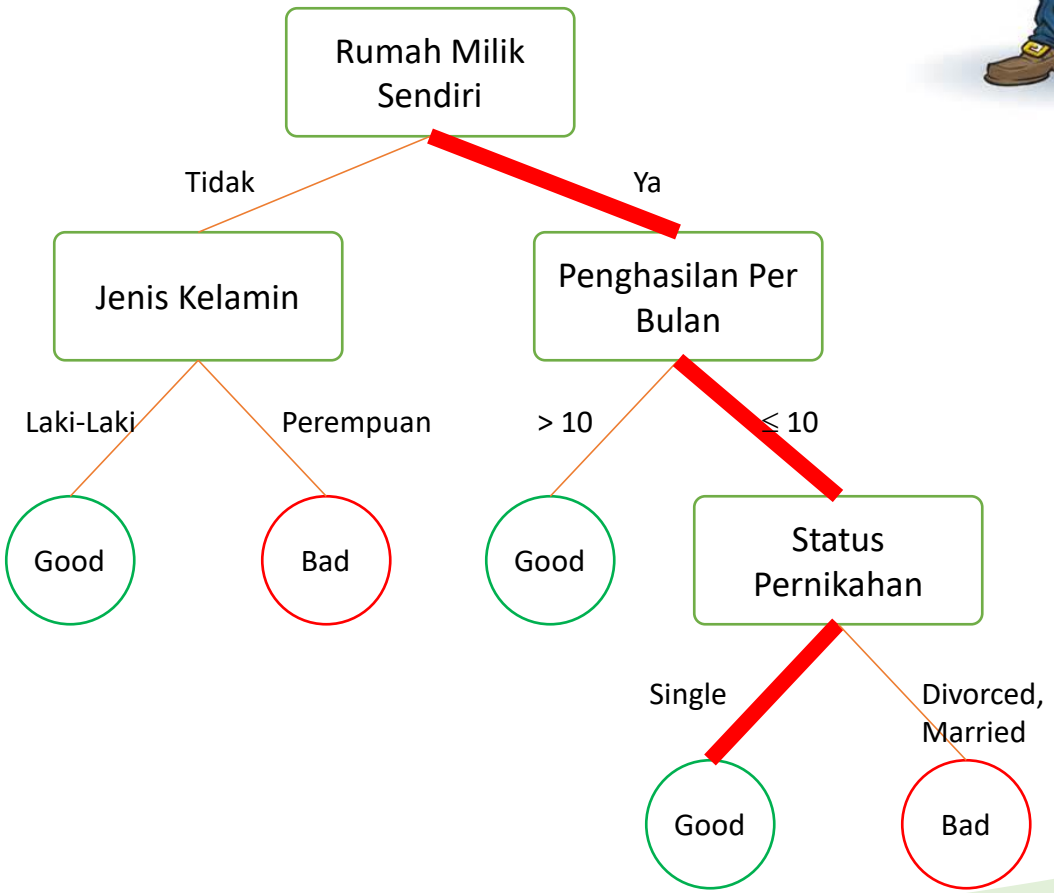
- Pria
- **Rumah Sendiri**
- **Penghasilan 8 juta per bulan**
- Bujangan



Ilustrasi penggunaan pohon klasifikasi

Profil:

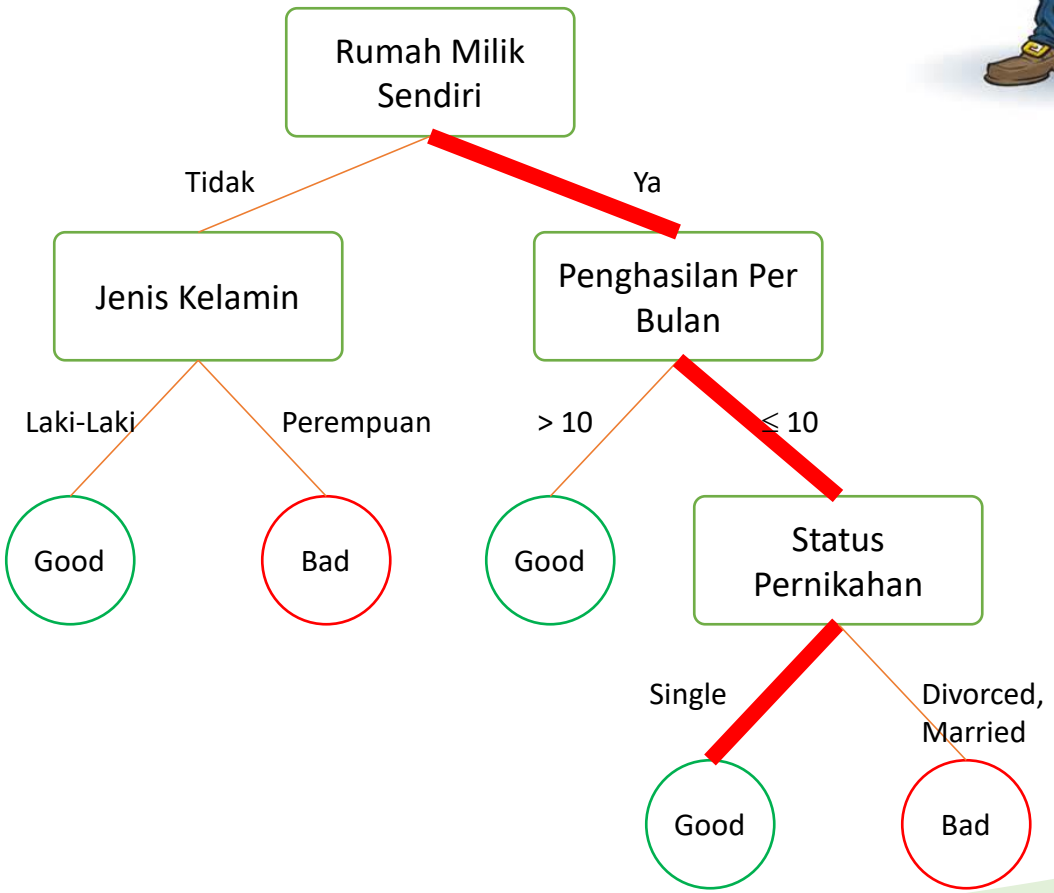
- Pria
- **Rumah Sendiri**
- **Penghasilan 8 juta per bulan**
- **Bujangan**



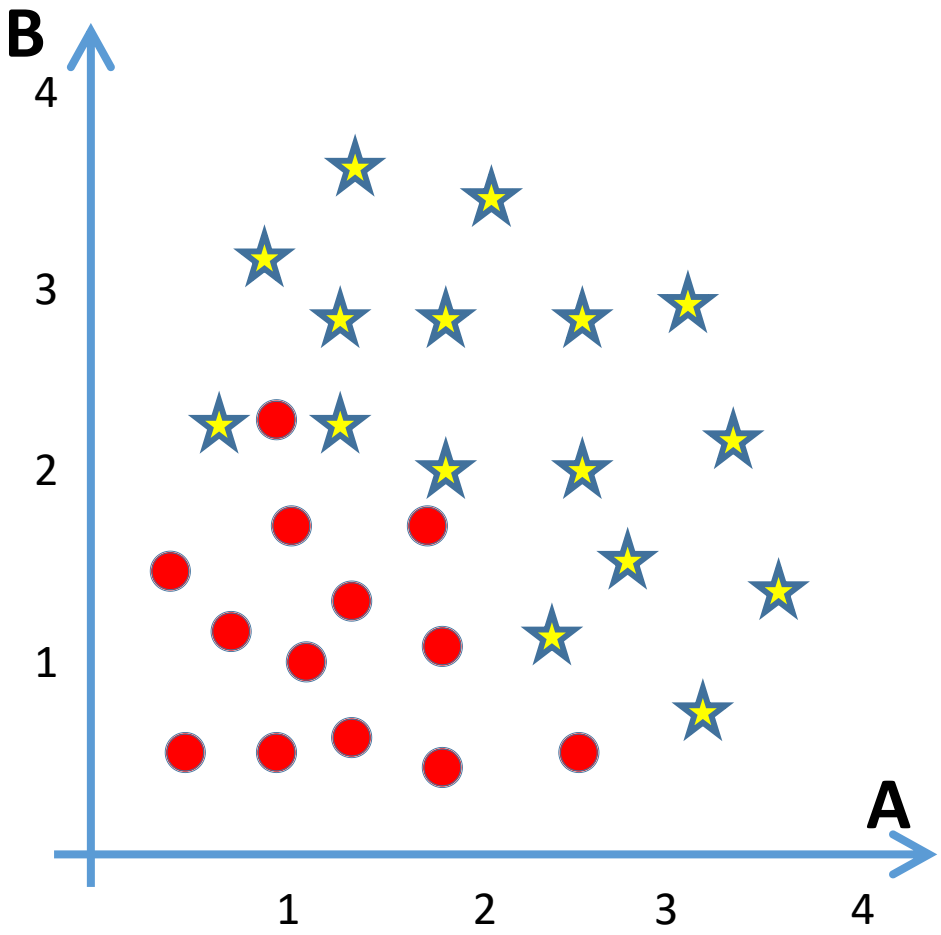
Ilustrasi penggunaan pohon klasifikasi

Profil:

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



Algoritma Pembentukan Pohon Klasifikasi



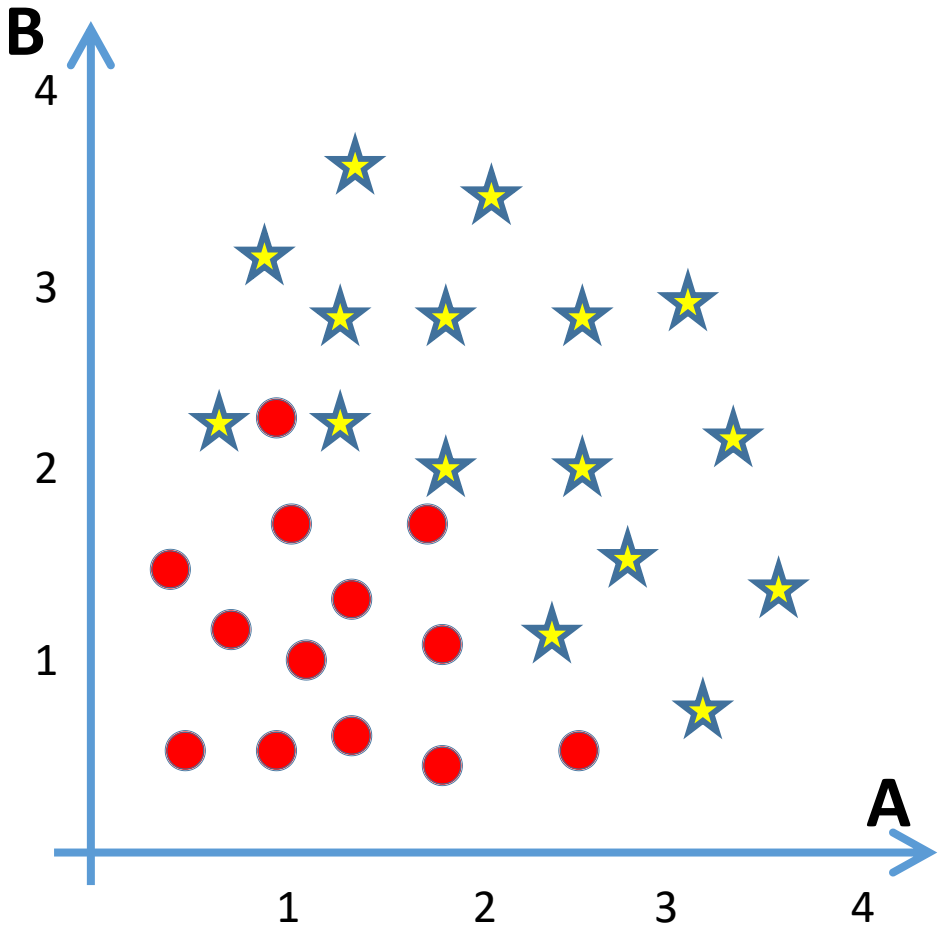
16 obs

13 obs

Mencari pemisah terbaik antara individu ★ dengan individu ●

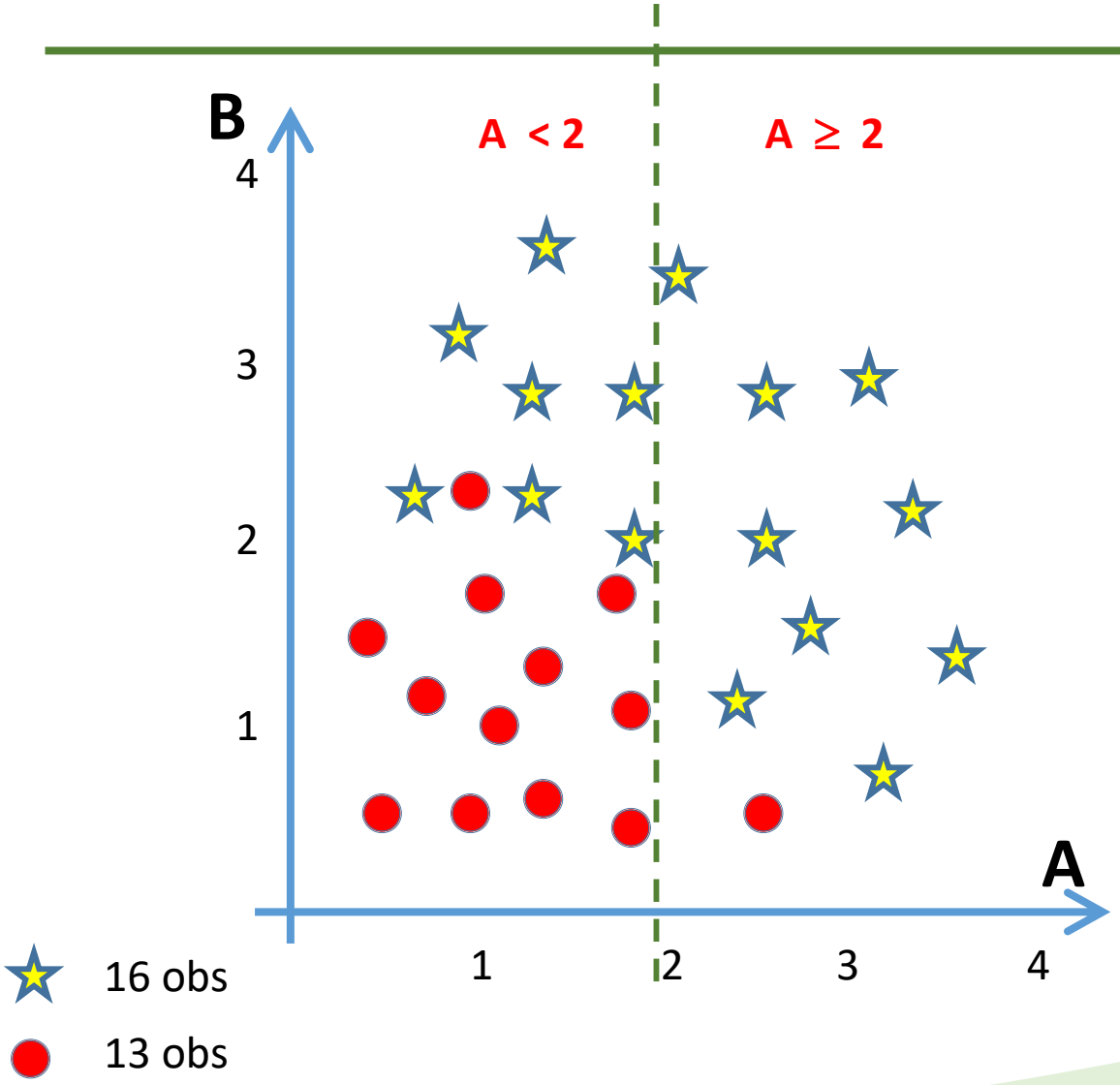
Pemisahan dilakukan untuk masing-masing variabel, bukan kombinasinya.

Algoritma Pembentukan Pohon Klasifikasi



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

Algoritma Pembentukan Pohon Klasifikasi



Pemisahan menggunakan garis A = 2, menghasilkan dua kelompok:

Kelompok 1 A < 2

7 obs

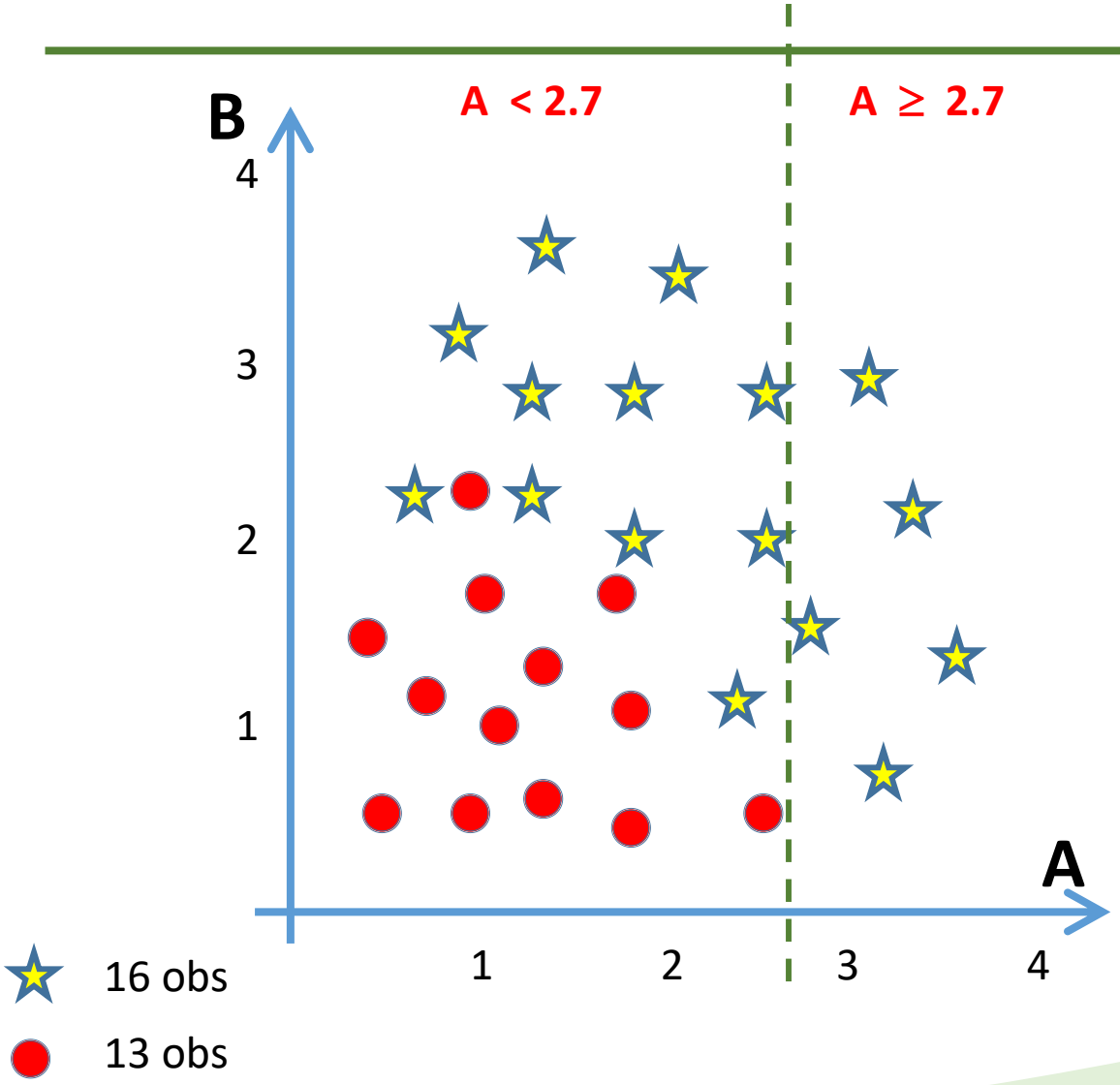
12 obs

Kelompok 2 A ≥ 2

9 obs

1 obs

Algoritma Pembentukan Pohon Klasifikasi



Pemisahan menggunakan garis A = 2.7, menghasilkan dua kelompok:

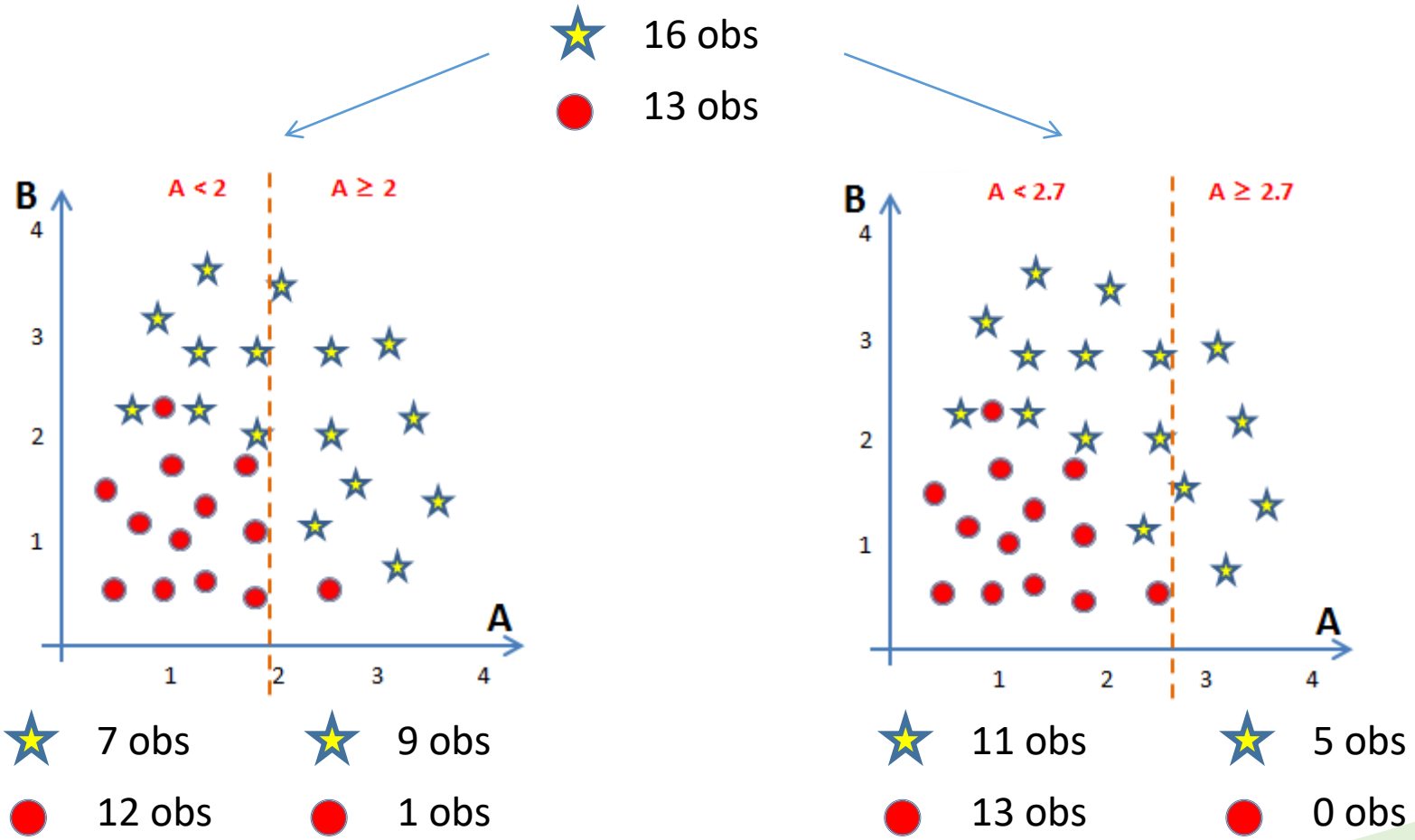
Kelompok 1 A < 2.7

- 11 obs
- 13 obs

Kelompok 2 A ≥ 2.7

- 5 obs
- 0 obs

Algoritma Pembentukan Pohon Klasifikasi



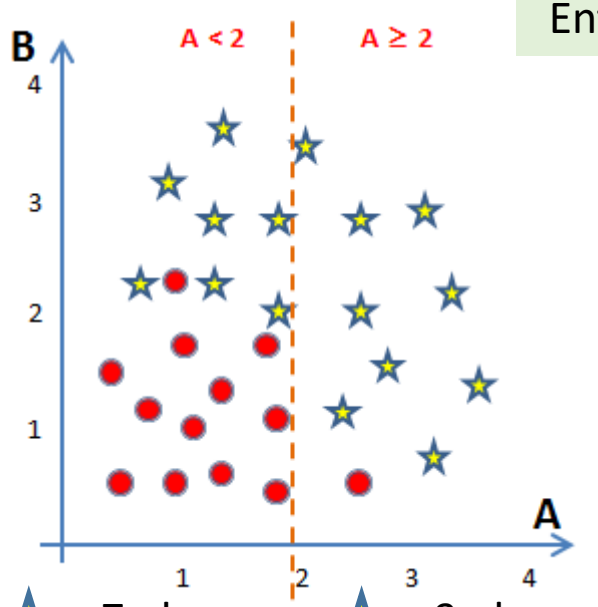
Mana yang lebih baik?

Algoritma Pembentukan Pohon Klasifikasi

16 obs

13 obs

Entropy = 0.9923



7 obs

12 obs

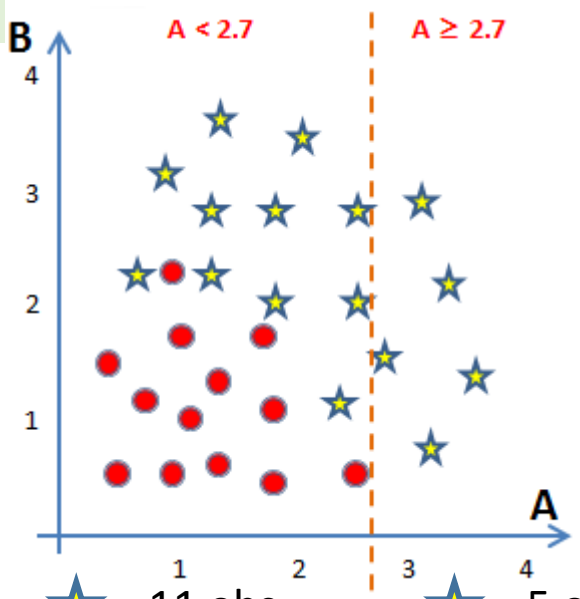
E = 0.9495

9 obs

1 obs

E = 0.4690

Information Gain = 0.2085



11 obs

13 obs

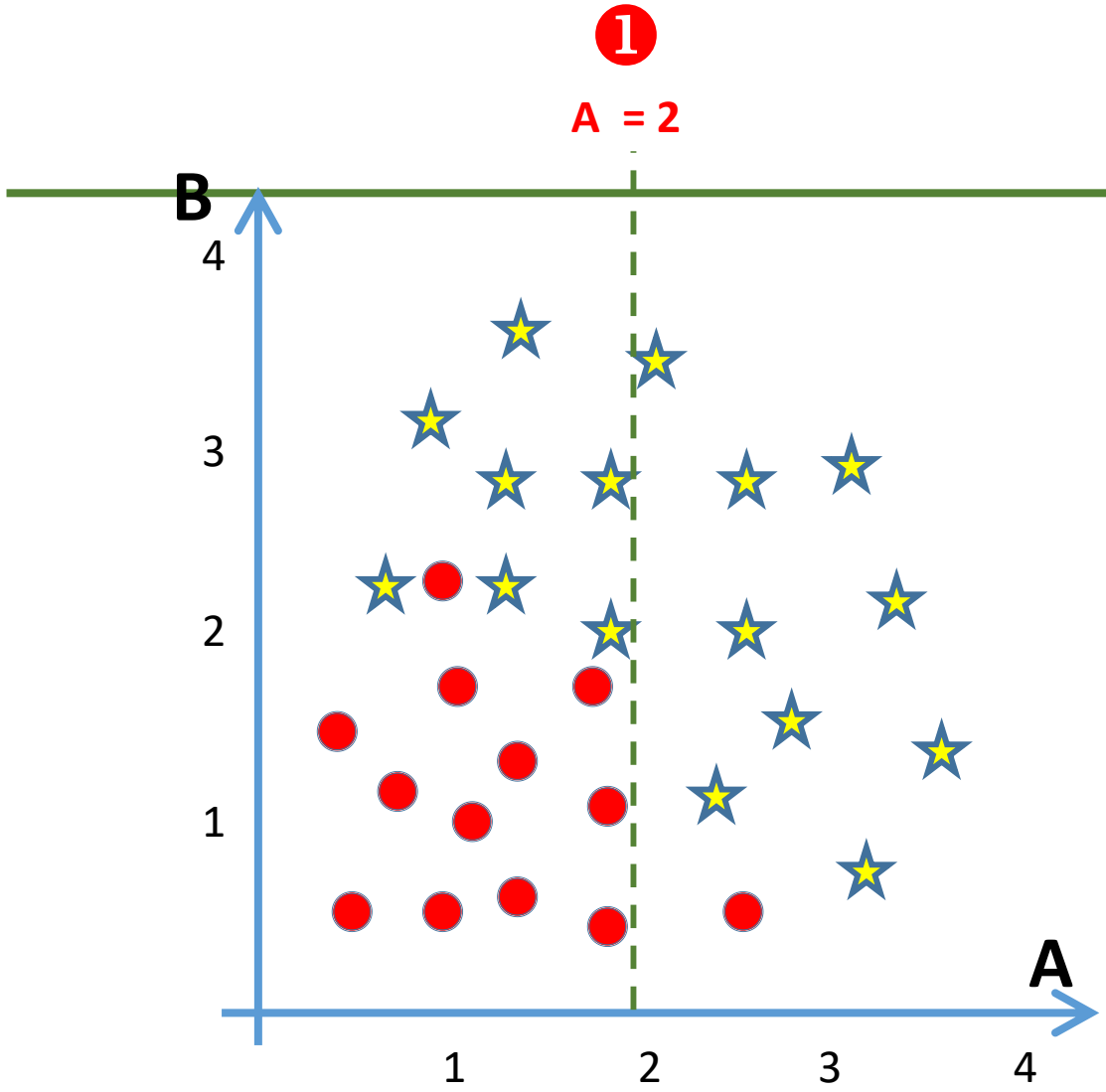
E = 0.9950

5 obs

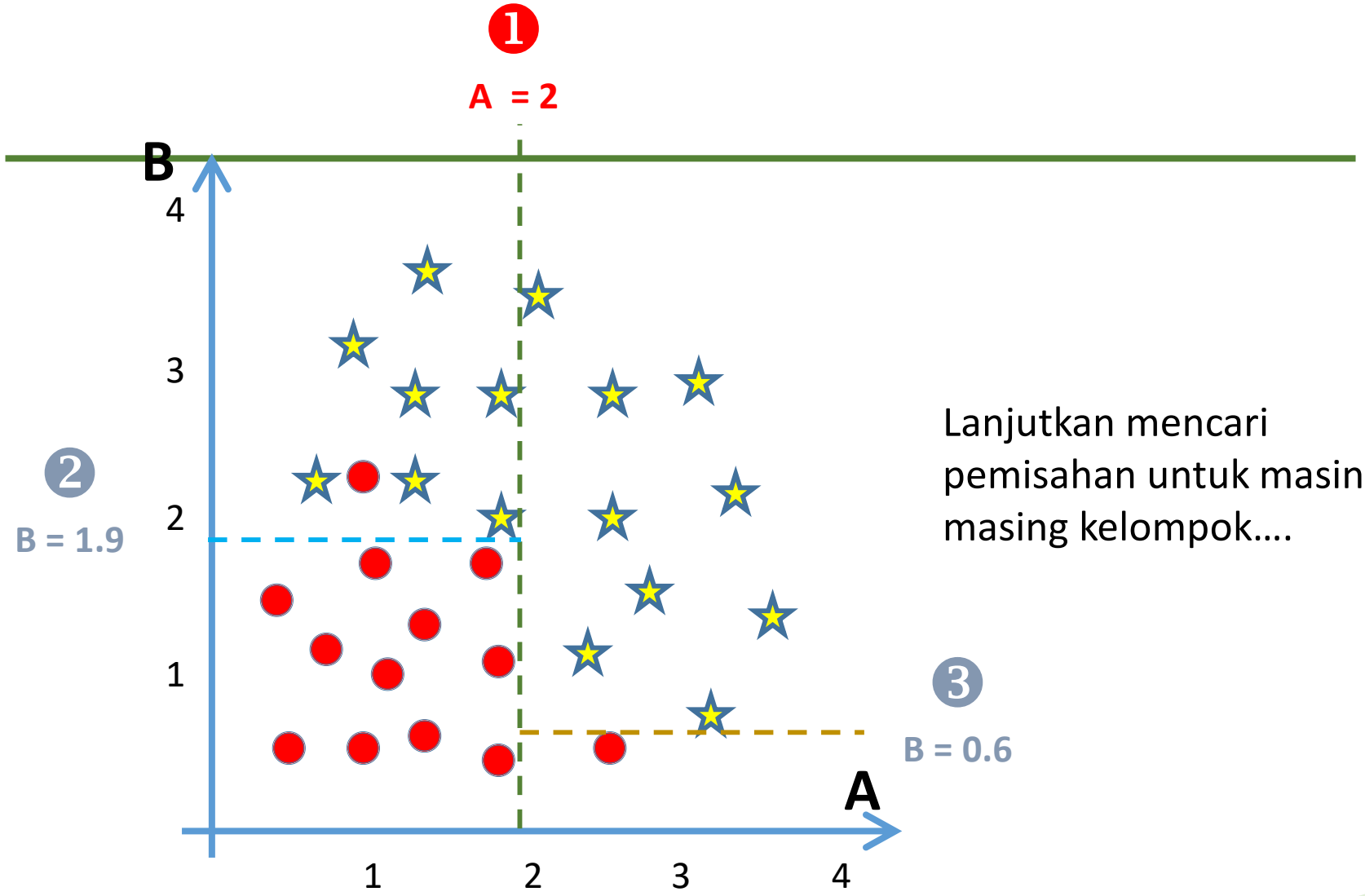
0 obs

E = 0

Information Gain = 0.1688

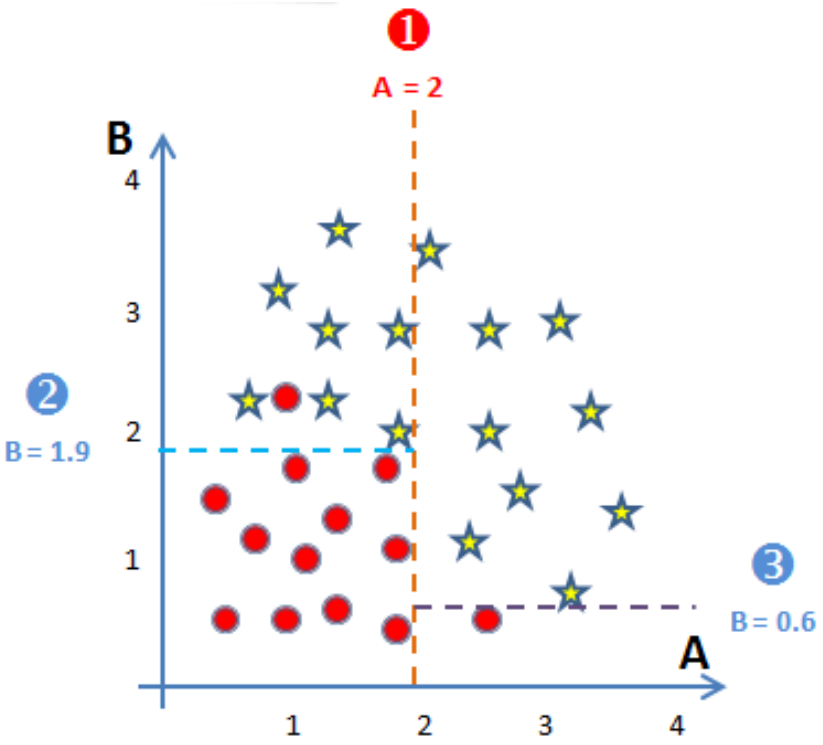


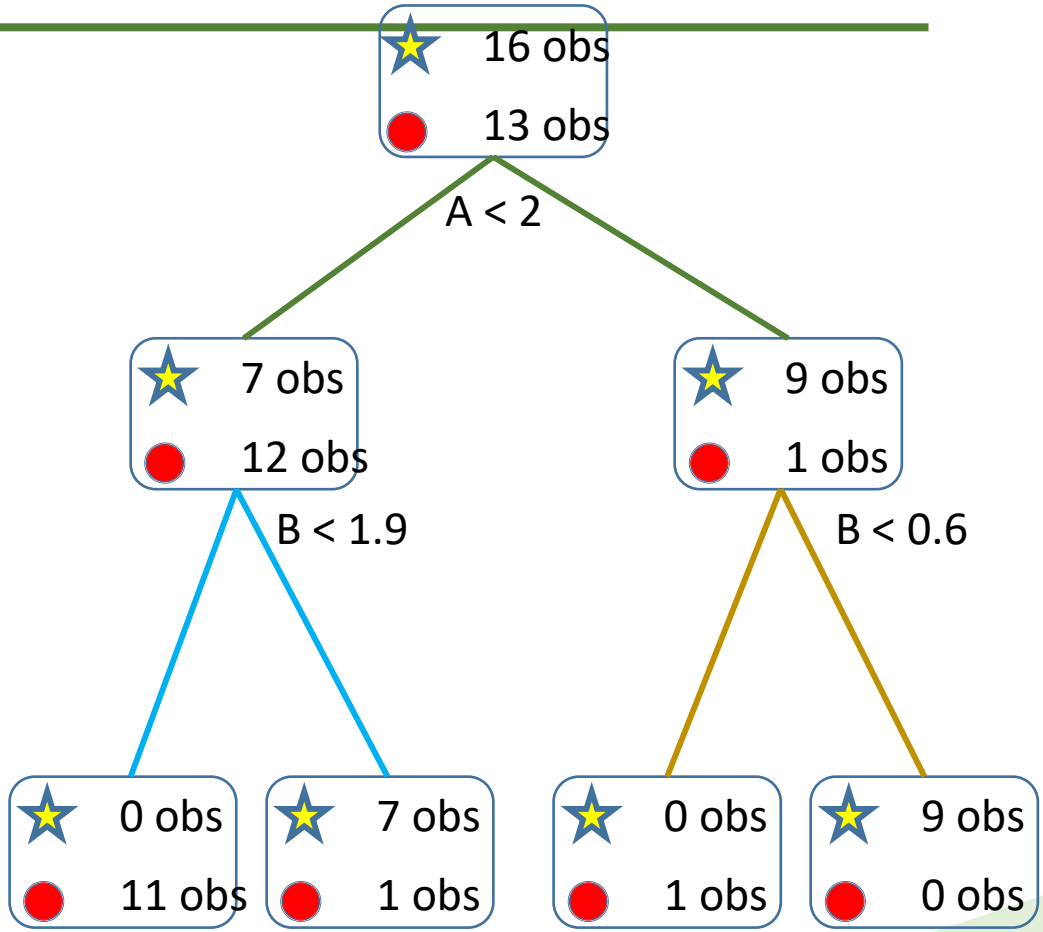
Lanjutkan mencari pemisahan untuk masing-masing kelompok….



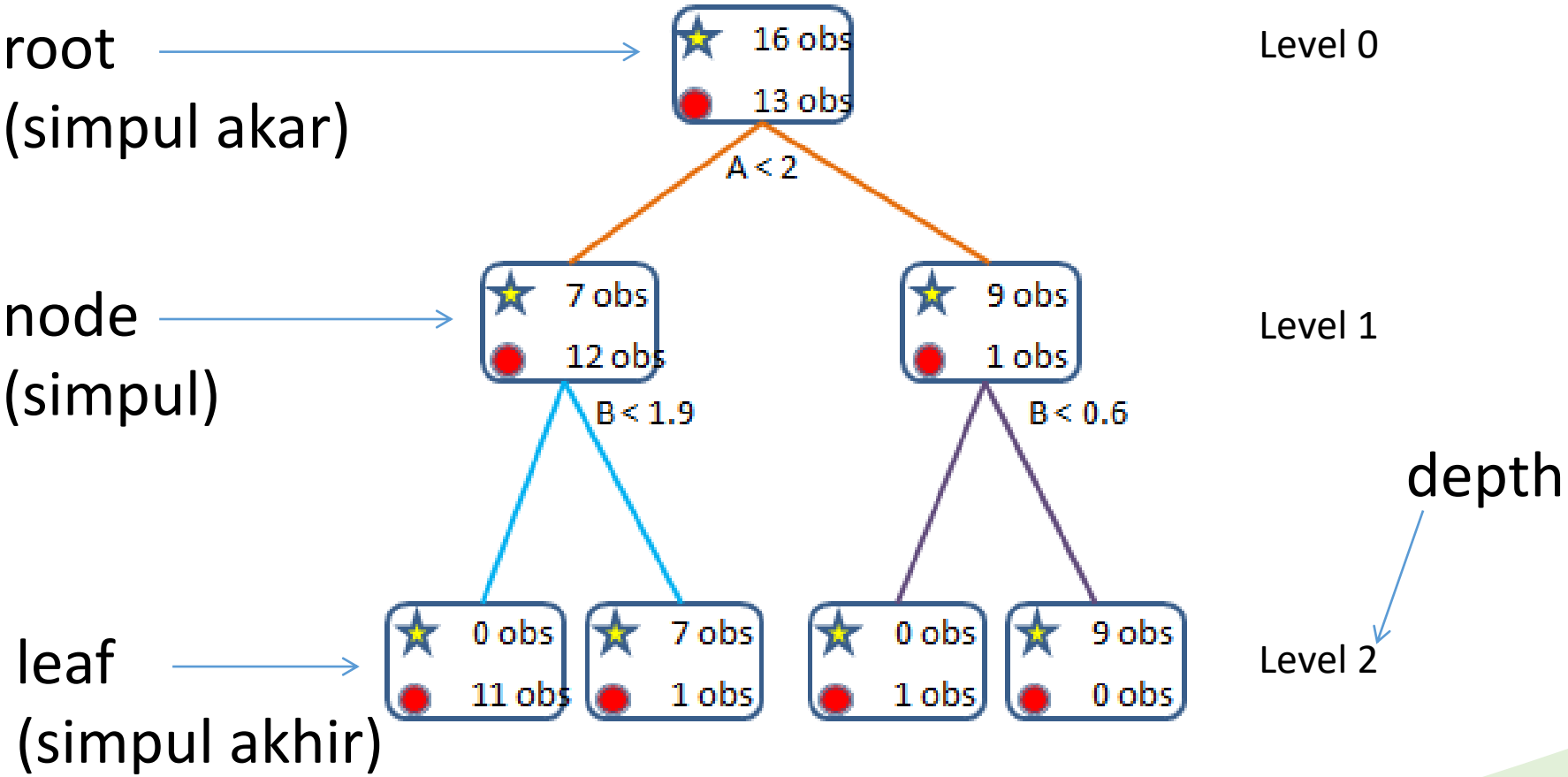
Lanjutkan mencari pemisahan untuk masing-masing kelompok….

Representasi Hasil Pemisahan





Pohon Klasifikasi



Algoritma Pembentukan Pohon Klasifikasi

- Tahap 1:
  Mencari pemisahan/penyekatan (splitting) terbaik di setiap variabel

- Tahap 2:
  Menentukan variabel terbaik untuk penyekatan

- Tahap 3:
  Melakukan penyekatan berdasarkan hasil dari Tahap 2, dan memeriksa apakah sudah waktunya menghentikan proses

Lakukan tiga tahapan di atas untuk setiap simpul dan hasil sekatannya

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: $C^*(x) = \underset{y}{\operatorname{argmax}} \sum_i \delta\big(C_i(x) = y\big)$.

$\{\delta(\cdot) = 1$ if its argument is true and 0 otherwise$\}$.

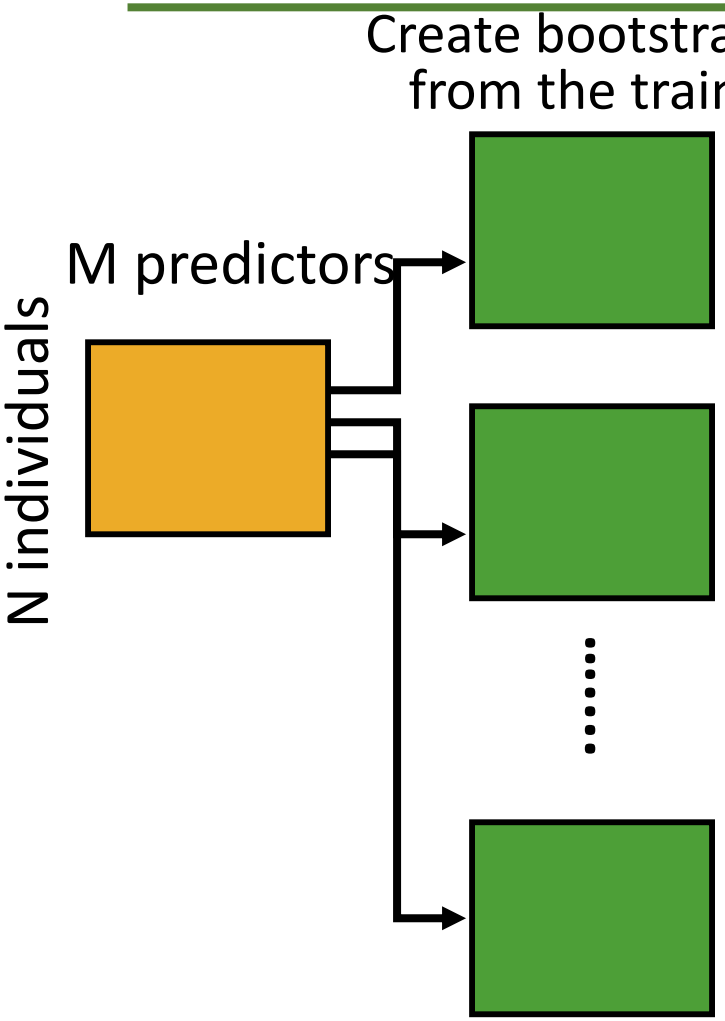
Bagging

Training Data

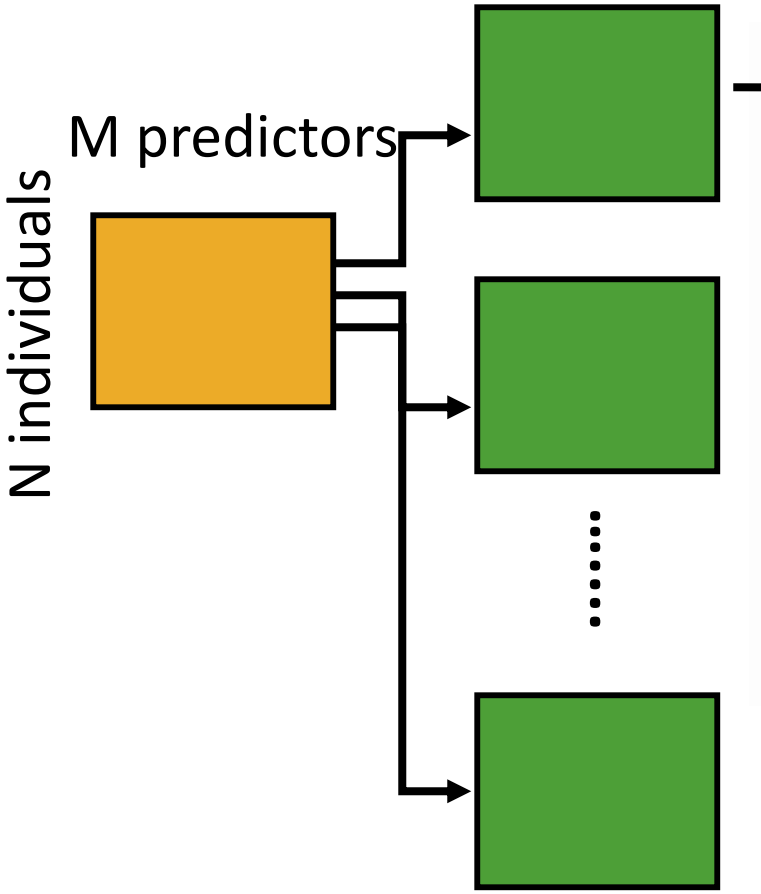
M predictors

N individuals

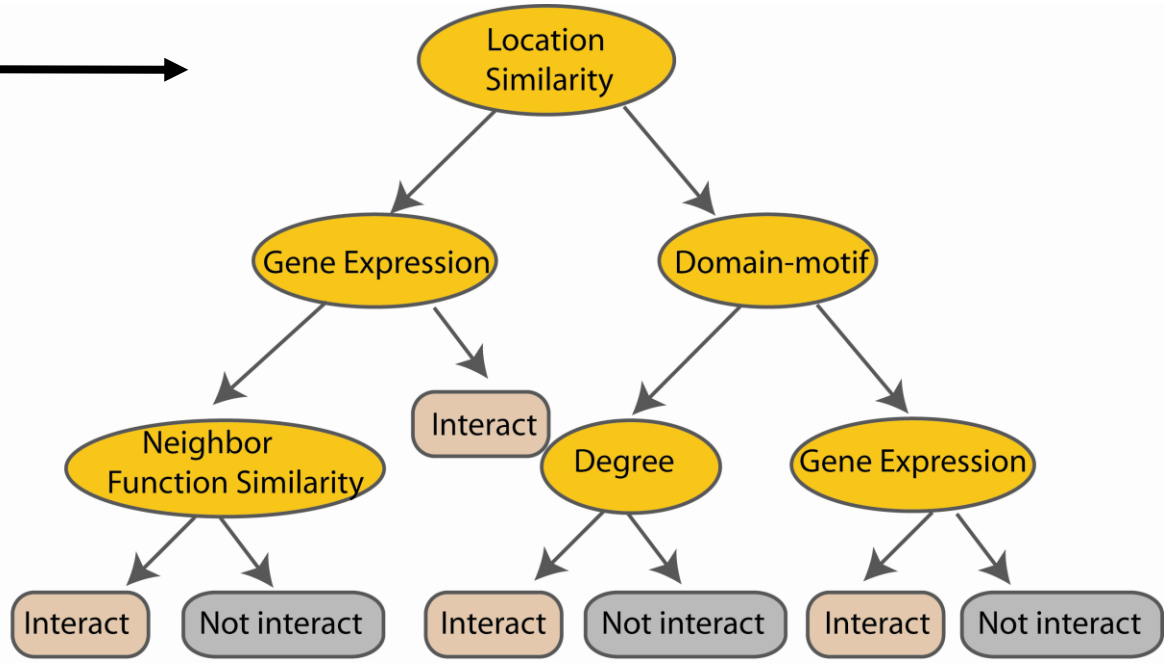
Bagging


Create bootstrap samples
from the training data
M predictors
N individuals

Bagging

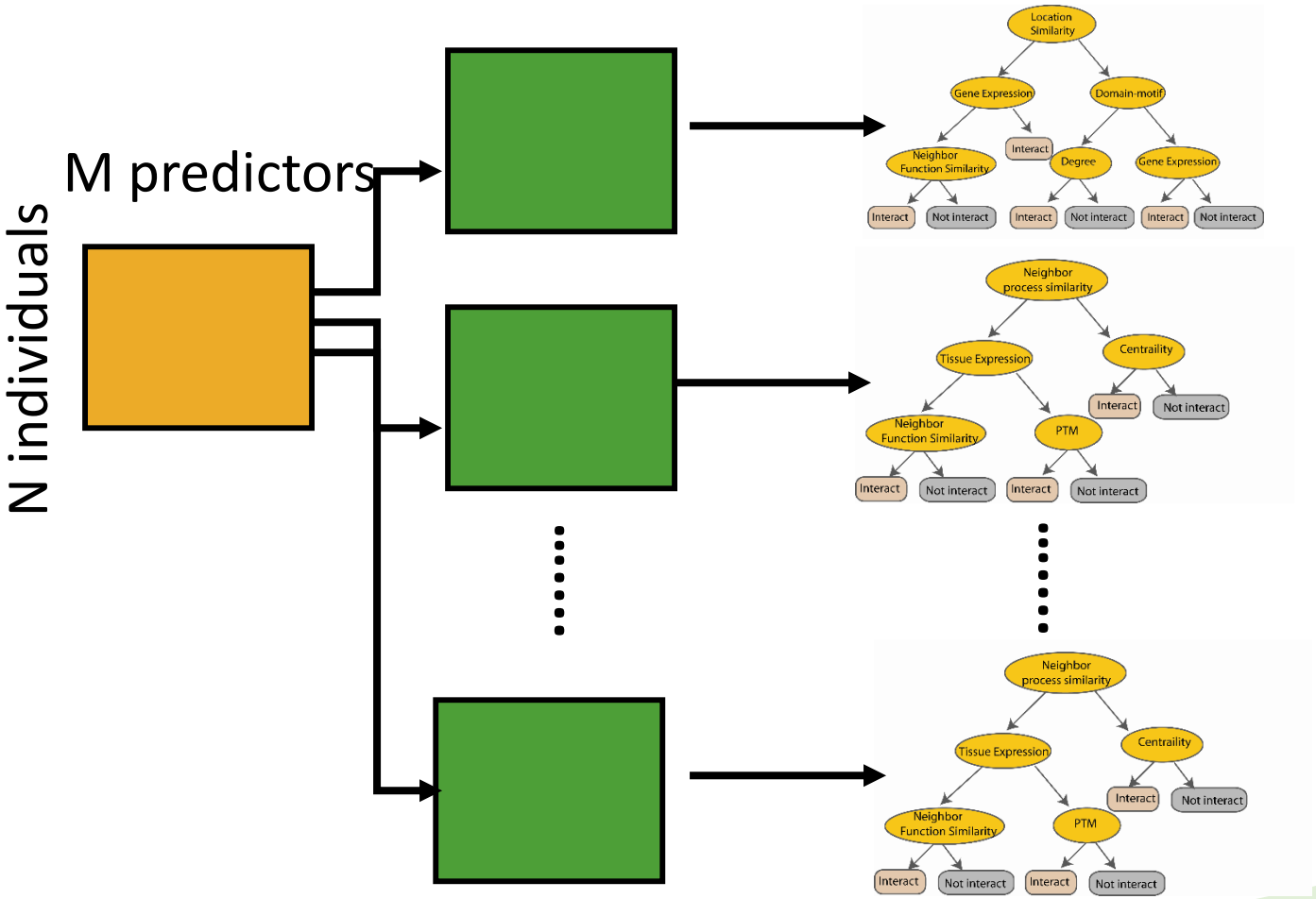


Construct a decision tree



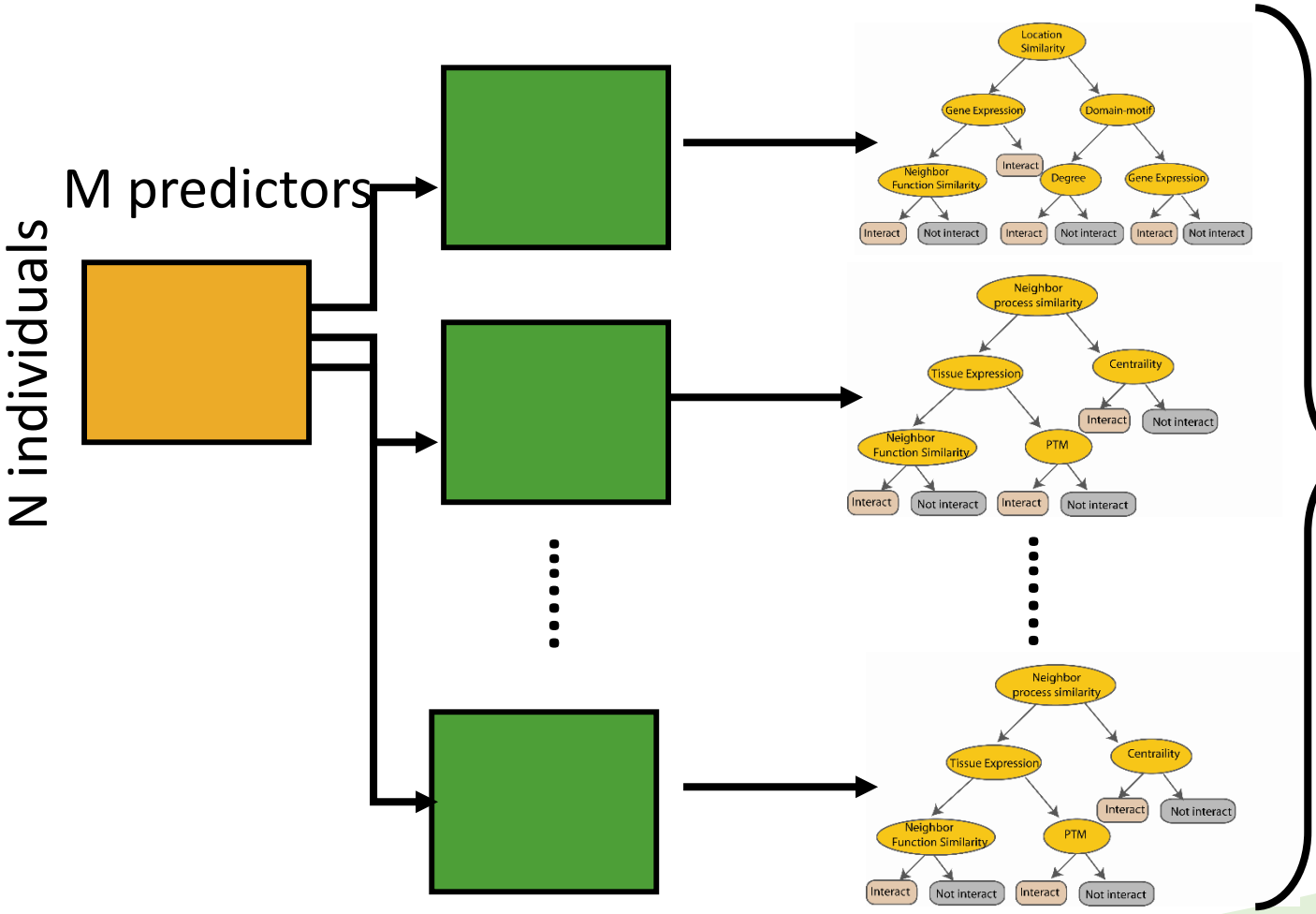
Bagging

Create decision tree from each bootstrap sample



Bagging

Create decision tree from each bootstrap sample



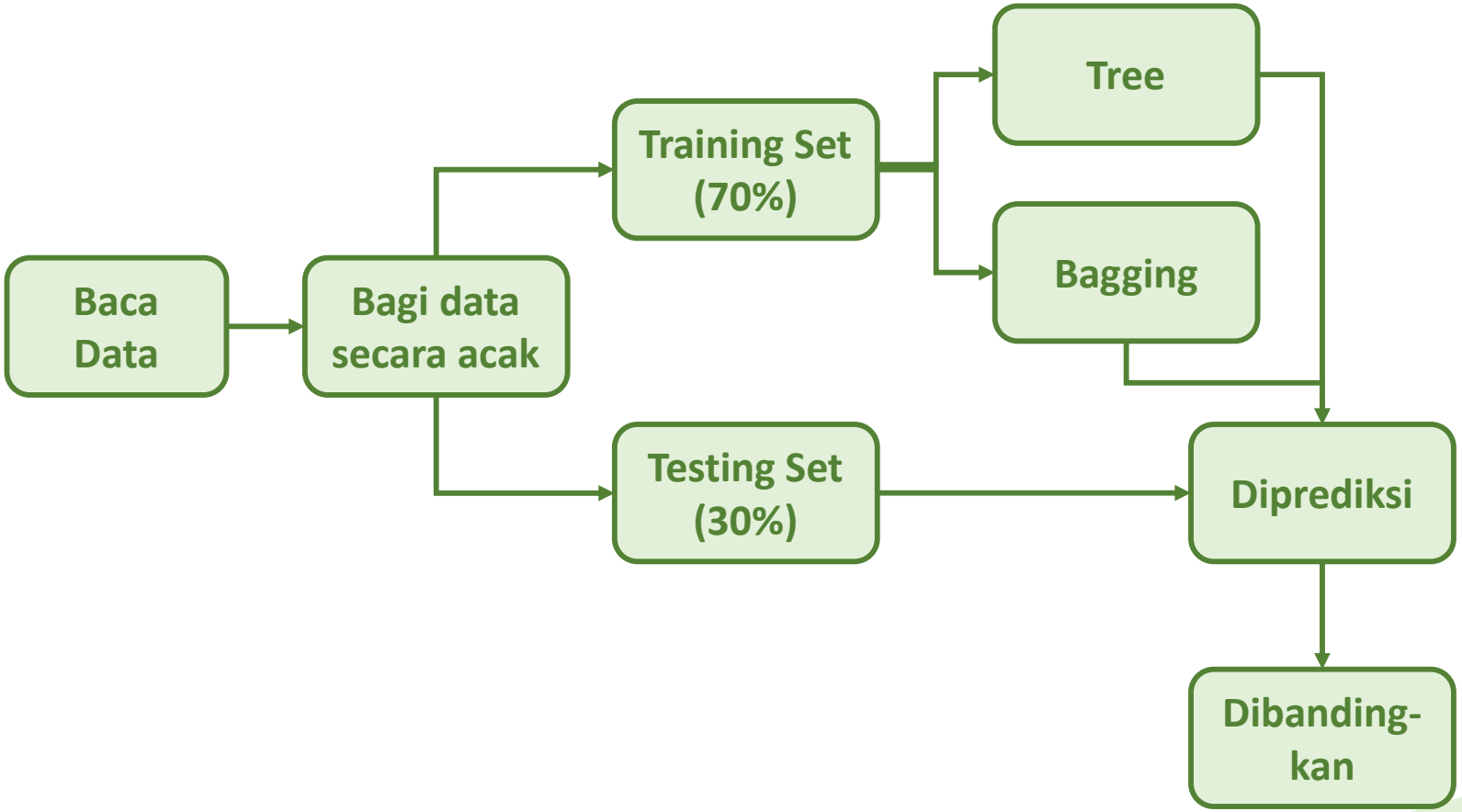
Take he majority vote

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 $\boldsymbol{n_{min}}$ 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:

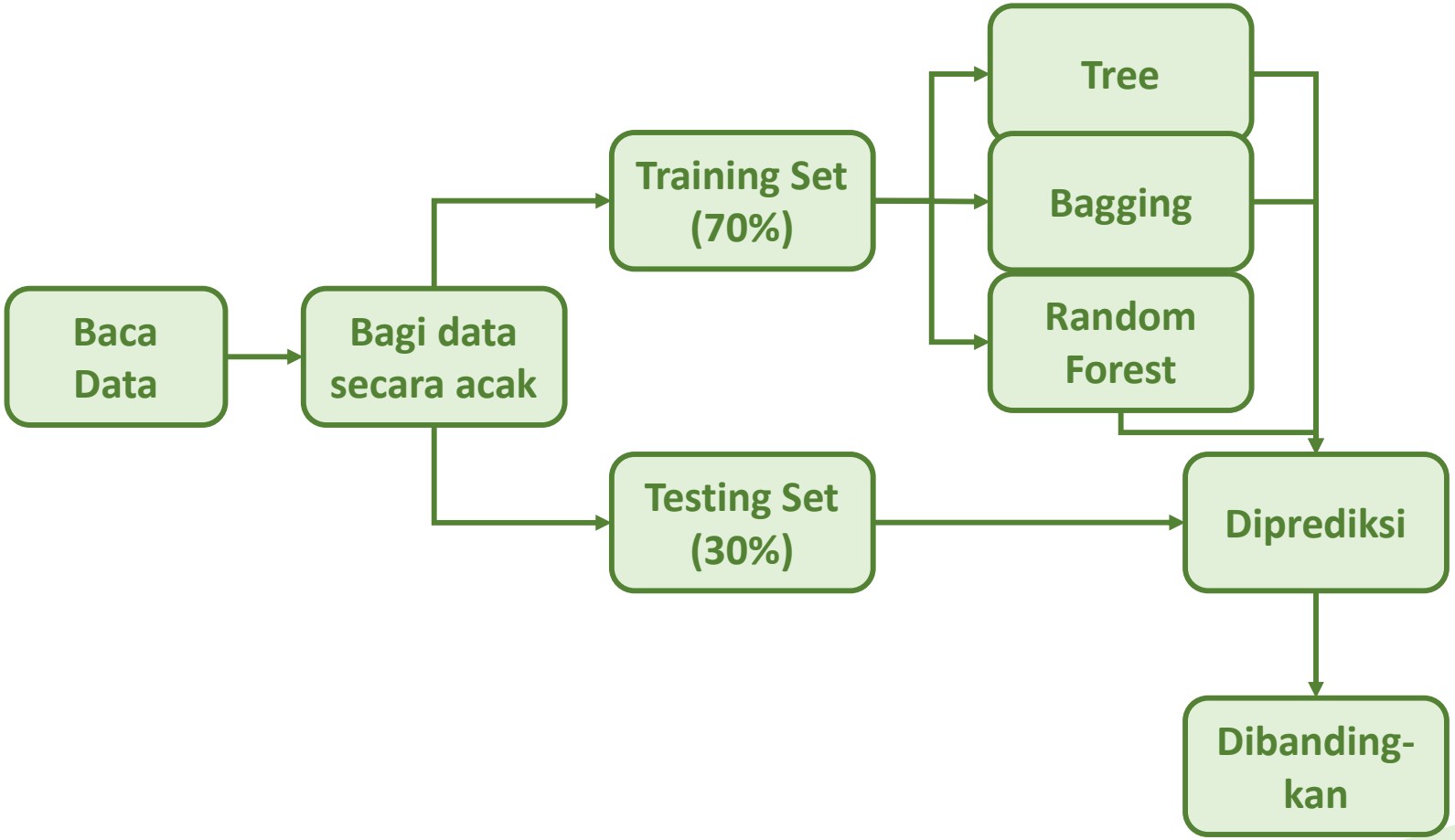
For regression: average the results

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 Nrecords 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

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

Iteration step (for t=1…T):

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

2. Compute the error rate ε_t as

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

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

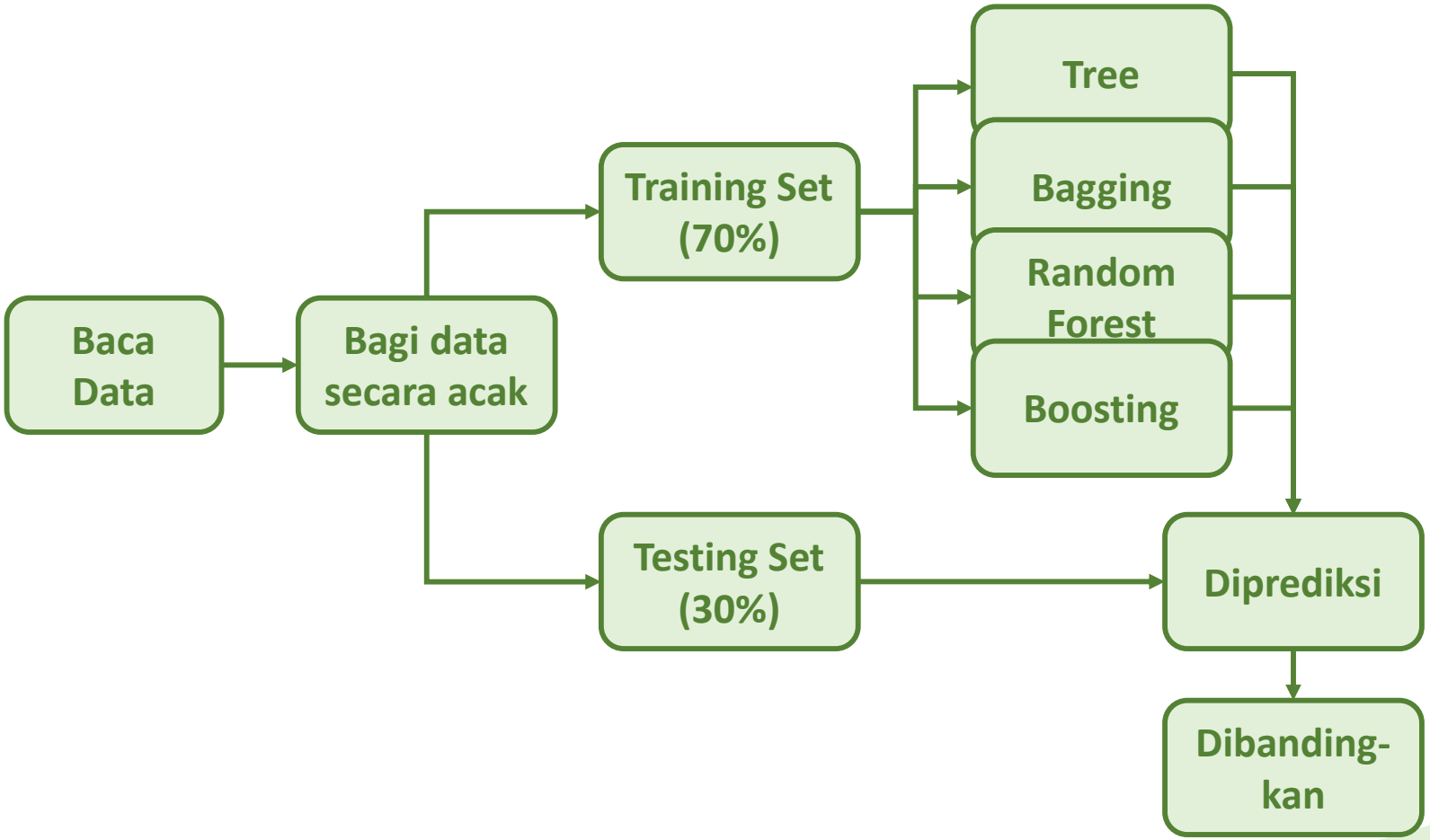
5.Normalize $D(x_i)$ so that $\sum_{i=1}^{N} D(x_i) = 1$

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

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.

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS

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*

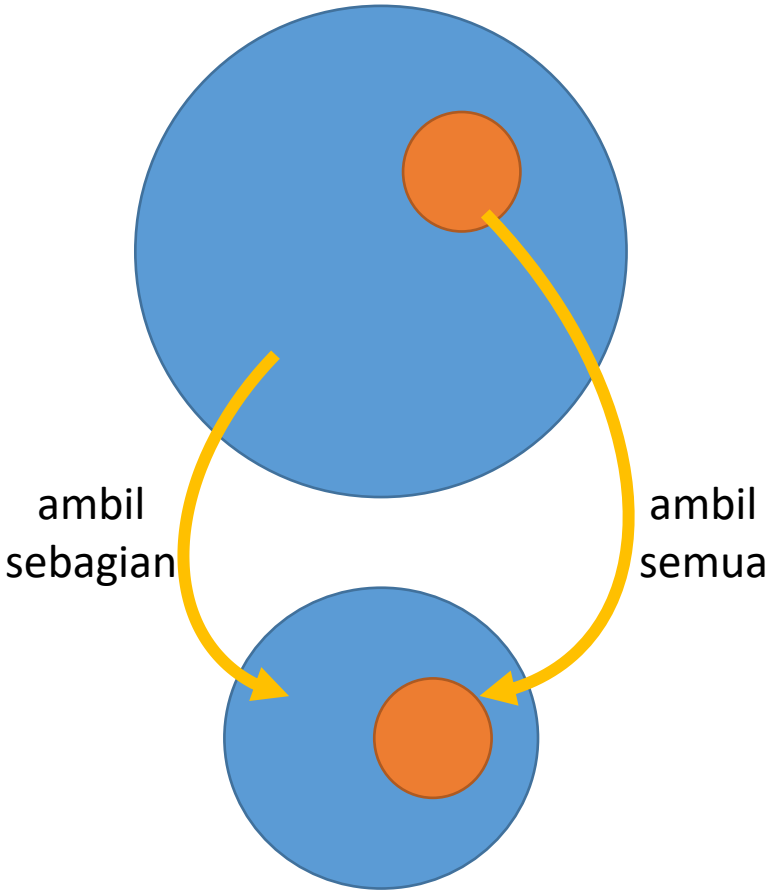
M. Galar, A. Fernandez, E. Barrenechea, H. Bustince and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463-484, July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

Beberapa pilihan

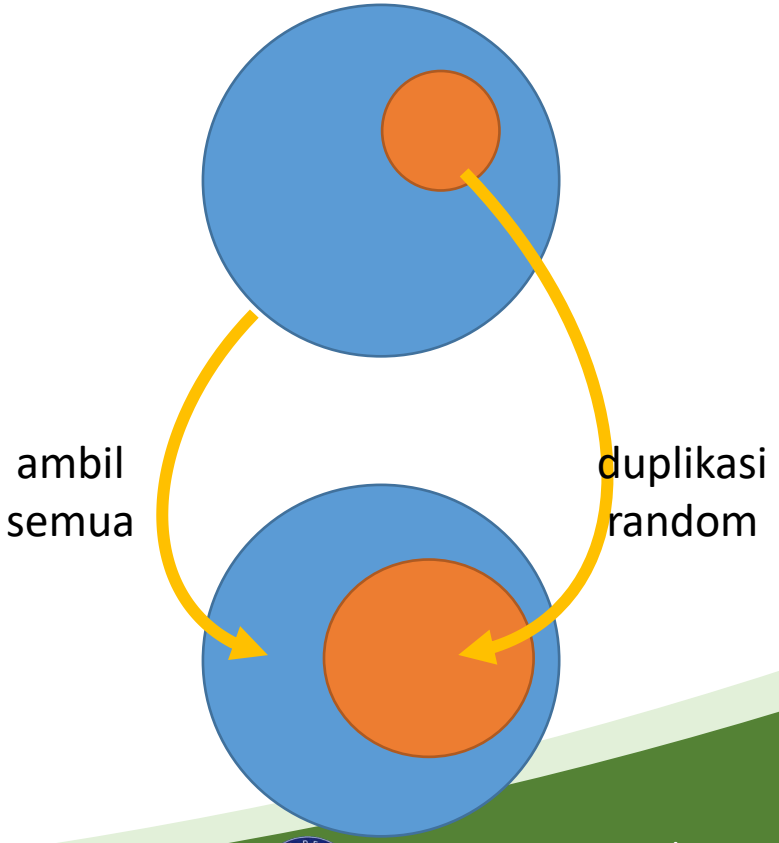
- Over-Bagging, gabungan antara oversampling dan Bagging
- Under-Sampling, gabungan antara undersampling dan Bagging
- EasyEnsemble , kombinasi undersampling dan Boosting
- RUSBoost, kombinasi undersampling dan Boosting
- dll

Undersampling dan Oversampling

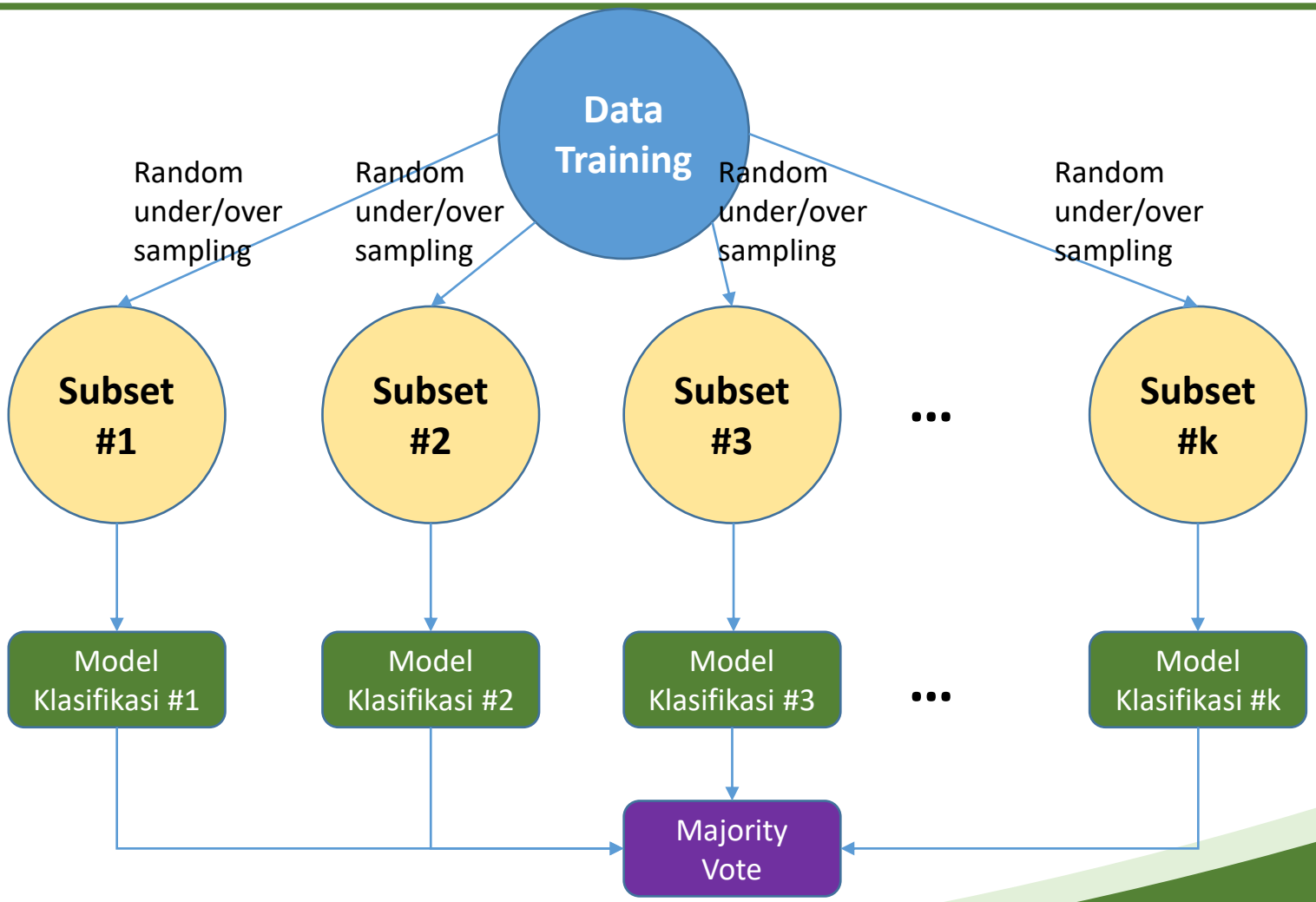
undersampling



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: $\quad i \Leftarrow i + 1$

5: $\quad$ Randomly sample a subset $\mathcal{N}_i$ from $\mathcal{N}$, $|\mathcal{N}_i| = |\mathcal{P}|$.

6: $\quad$ 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: $H(x) = \text{sgn}\left(\sum_{i=1}^{T} \sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x) - \sum_{i=1}^{T} \theta_i\right)$.

Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* (2), 539-550.

RUS-Boost

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 $(x_1, y_1), ..., (x_m, y_m)$ 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 $t = 1, 2, ..., T$

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:

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

Mari kita coba di R

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

Terima Kasih