



Penyiapan Data

Kuliah 2 – STA1382 Teknik
Pembelajaran Mesin

Septian Rahardiantoro



Materi Perkuliahan

No	Materi
1	Ruang Lingkup Pembelajaran Mesin
2	Penyiapan Data
3	Regresi Linier dan Regresi Logistik serta Evaluasi
4	Metode Berbasis Pohon (CART) dan Evaluasi
5	Neural Network
6	Support Vector Machine
7	Studi Kasus (Presentasi)
UTS	

No	Materi
8	Penggerombolan Berhierarki dan Evaluasi
9	Penggerombolan Non-hierarki dan Evaluasi
10	Reduksi Dimensi dengan PCA
11	Association Rule
12	Metode Ensemble
13	Kasus: Dosen Tamu
14	Studi Kasus (Makalah)
UAS	

Rencana Perkuliahan UTS

Kuliah : setiap Hari **Senin pukul 10:00 WIB**
RK. U 3.03

Praktikum : setiap Hari **Senin pukul 16:00 WIB**

Tanggal	Materi	Pelaksanaan	Keterangan
30 Januari 2023	Penyiapan Data	Offline	
6 Februari 2023	Regresi Linier dan Regresi Logistik beserta evaluasinya	Offline	Tugas Kelompok*
13 Februari 2023	Metode berbasis pohon (CART) dan evaluasinya	Online	
20 Februari 2023	Neural Network	Offline	Tugas Individu*
27 Februari 2023	Support Vector Machine	Online	Kuis*
6 Maret 2023	Studi Kasus	Offline	Presentasi Tugas Kelompok*

*teknis dan waktunya akan dijelaskan lebih lanjut

Komponen Penilaian

Komponen penilaian:

Project	: 50%
- Kelompok (Presentasi UTS)	: 15%
- Kelompok (Paper)	: 20%
- Individu	: 15%
UTS	: 20%
UAS	: 20%
Praktikum	: 10%

Outline

- Data Hilang dan Metode mengatasinya
- Pencilan dan Metode penanganannya
- Metode transformasi data

Data Hilang

- Data hilang (missing value) → merupakan suatu kejadian ketika tidak ada nilai data yang terekam/ tercatat pada pengamatan untuk suatu variabel
- Penyebab data hilang:
 - Non-respon di survei: pertanyaan sensitif, tidak tahu jawaban, pengukuran berulang
 - Kondisi pengamatan rusak
 - Kesalahan peneliti atau entri data

- Analisis statistika yang tetap digunakan pada data hilang dapat menyebabkan kesimpulan yang diperoleh tidak merepresentasikan kondisi populasi yang sebenarnya.
- Selain itu, alasan mengapa terdapat data hilang penting diketahui untuk dapat menangani data yang tersisa dengan benar.
- Jika nilai hilang secara sistematis, analisis yang dilakukan mungkin akan bias.

Jenis Mekanisme Data Hilang

1. Missing Completely at Random (MCAR).

- Jika peristiwa yang menyebabkan hilangnya nilai data tertentu tidak bergantung pada variabel yang dapat diamati dan yang tidak dapat diamati, dan terjadi seluruhnya secara acak. Serta hilangnya nilai variabel ini tidak ada hubungannya dengan objek yang sedang dipelajari.
- Oleh karena itu, data hilang dan data lengkap tidak memberikan hasil berbeda saat analisis. Sehingga, kasus ini dapat dikatakan hilang secara acak penuh dari data.

Jenis Mekanisme Data Hilang (2)

2. Missing at Random (MAR).

- Data hilang tergantung dari nilai yang diketahui, dengan demikian dapat dideskripsikan oleh peubah penjelas dalam data set. Menghitung nilai “penyebab” data hilang akan menghasilkan hasil tak bias dalam analisis.

3. Missing Not at Random (MNAR).

- Data hilang disebabkan oleh kejadian atau materi yang tidak diukur. Nilai variabel yang hilang terkait dengan alasan hilangnya variabel tersebut.

- Jika data hilang termasuk dalam kategori MCAR, analisis yang dilakukan pada data tidak bias meskipun data hilang tersebut tidak ditangani. Namun, kondisi data hilang MCAR sangat jarang untuk ditemui. Dalam kasus MCAR, data pengamatan yang hilang tidak terkait dengan variabel apa pun: dengan demikian, pengamatan dengan data yang benar-benar diamati pada dasarnya adalah sampel acak dari populasinya.
- Jika data hilang termasuk dalam kategori MAR, data yang terdapat data hilang tersebut akan berbias jika langsung dilakukan analisis. Alangkah lebih baik jika pada nilai data hilang tersebut diduga terlebih dahulu menggunakan variabel-variabel lainnya yang lengkap sebelum melakukan analisis.
- Jika data hilang termasuk dalam kategori MNAR, kondisi ini sangat sulit untuk ditangani karena hilangnya data berdasarkan alasan diluar kendali penelitian. Sehingga perlu ditelusuri penyebab hilangnya nilai data tersebut, serta penanganan yang tepat ketika hal ini terjadi.

Ilustrasi Kasus Data Hilang (1)

- Tidak terisinya beberapa pengamatan variabel jenis kelamin pada responden suatu survei. Sedangkan variabel jenis kelamin ini tidak bergantung pada variabel lainnya dalam survei tersebut.
- **Dalam hal ini data hilang sepenuhnya secara acak (MCAR).**

Ilustrasi Kasus Data Hilang (2)

- Misalnya tidak terisinya variabel tingkat pendapatan yang notabene nya dapat diduga dari variabel lainnya, seperti jenis pekerjaan, tingkat pendidikan, usia, dan lainnya.
- **Kondisi ini merupakan ilustrasi kasus MAR**, karena data hilang pada variabel tertentu dapat diduga berdasarkan data lengkap pada variabel lainnya.

Illustrasi Kasus Data Hilang (3)

- Pada survei mengenai depresi, mekanisme MNAR dapat terjadi jika seseorang gagal mengisi surveinya karena tingkat depresi yang dideritanya.
- **Data tersebut tidak hilang secara sembarangan (MNAR)**

Implikasi Data Hilang dalam Analisis

1. ketiadaan data pengamatan akan mengurangi kekuatan statistik, uji yang dihasilkan akan cenderung untuk tidak tolak hipotesis nol
2. adanya data hilang dapat meningkatkan galat baku (standard error)
3. data yang hilang dapat menyebabkan bias dalam pendugaan parameter
4. dapat mengurangi keterwakilan sampel sehingga melemahkan generalisasi dari hasil
5. adanya data hilang dapat mempersulit analisis penelitian, karena sebagian besar metode-metode analisis statistika dirancang untuk data lengkap

Metode Mengatasi Data Hilang (Teknik Sederhana)

1. Penghapusan Kasus (Listwise or Case Deletion)

- dengan menghilangkan/ menghapus kasus pengamatan sepenuhnya jika terdapat data hilang pada salah satu variabel dalam analisis.
- metode penghapusan kasus dapat digunakan sebagai alternatif penanganan data hilang yang cukup optimal apabila asumsi MCAR terpenuhi dengan cukup banyak sampel yang tersedia

2. Penghapusan Berpasangan (Pairwise Deletion)

- penghapusan pengamatan data hilang hanya dilakukan pada variabel yang akan dianalisis, yang kemudian analisis data dapat diselesaikan pada subset data tergantung nilai pengamatan mana yang hilang.
- pendekatan ini disarankan untuk diterapkan pada data hilang dengan mekanisme MCAR dan MAR.

Metode Mengatasi Data Hilang (Teknik Sederhana) (2)

3. Penggantian dengan Rataan (Mean Substitution)

- pendekatan ini mengganti nilai yang hilang pada variabel tertentu dengan nilai rataan sampelnya, sehingga menjadi seolah-olah data lengkap
- pendekatan ini cukup baik diterapkan pada data hilang dengan mekanisme MCAR dan MAR

4. Pengisian dengan Regresi (Regression Imputation)

- Pendekatan ini mempertahankan semua kasus pengamatan dengan mengganti nilai data yang hilang dengan nilai kemungkinan yang diduga oleh informasi variabel lain yang tersedia melalui pemodelan regresi
- Pendekatan ini baik digunakan pada data hilang dengan mekanisme MAR.

Ilustrasi

- Diketahui data karakteristik ikan, yang terdiri dari 3 peubah: tinggi, lebar, dan panjang ikan.

```
dataikan <- read.csv("data ikan.csv", header=T)
str(dataikan)
summary(dataikan)
> str(dataikan)
'data.frame': 55 obs. of 3 variables:
$ Panjang: num 30 31.2 31.1 33.5 34 34.7 34.5 35 35.1 36.2 ...
$ Tinggi : num 11.5 12.5 12.4 12.7 12.4 ...
$ Lebar   : num 4.02 4.31 4.7 NA 5.13 ...
> summary(dataikan)
Panjang          Tinggi          Lebar
Min.   :16.20    Min.   : 4.147    Min.   :2.268
1st Qu.:26.75   1st Qu.: 7.060    1st Qu.:3.820
Median :35.00   Median :13.759    Median :4.927
Mean   :33.49   Mean   :12.097    Mean   :4.781
3rd Qu.:39.35   3rd Qu.:15.496    3rd Qu.:5.580
Max.   :46.50   Max.   :18.957    Max.   :6.750
NA's    :2
```

Akan dilakukan analisis regresi:
Model 1: Tinggi ~ Panjang + Lebar
Model 2: Tinggi ~ Panjang

- Penanganan dengan Teknik Sederhana

```
##penghapusan kasus (listwise or case deletion)
datahilang <- dataikan[is.na(dataikan$Lebar), ]
datahilang
datalengkap <- dataikan[!is.na(dataikan$Lebar), ]
head(datalengkap)
model.1 <- lm(Tinggi~.,data=datalengkap)
model.1
model.2 <- lm(Tinggi~Panjang,data=datalengkap)
model.2
```

Model 1 dan model 2 menggunakan data lengkap tanpa pengamatan ke-4 dan ke-23

```
##penghapusan berpasangan (pairwise deletion)
model.1 <- lm(Tinggi~.,data=datalengkap)
model.1
model.2 <- lm(Tinggi~Panjang,data=dataikan)
model.2
```

Model 1 menggunakan data lengkap tanpa pengamatan ke-4 dan ke-23

Model 2 menggunakan data awal

```
##penggantian dengan rataan (mean substitution)
rataan.lebar <- mean(dataikan$Lebar)
rataan.lebar
Lebar.2 <- ifelse(is.na(dataikan$Lebar),rataan.lebar,dataikan$Lebar)
datalengkap.2 <- data.frame(dataikan[,1:2],Lebar.2)
head(datalengkap.2)
model.1 <- lm(Tinggi~.,data=datalengkap.2)
model.1
model.2 <- lm(Tinggi~Panjang,data=datalengkap.2)
model.2
```

Model 1 dan model 2 menggunakan data lengkap dengan pengamatan ke-4 dan ke-23 diganti dengan nilai rataan peubahnya

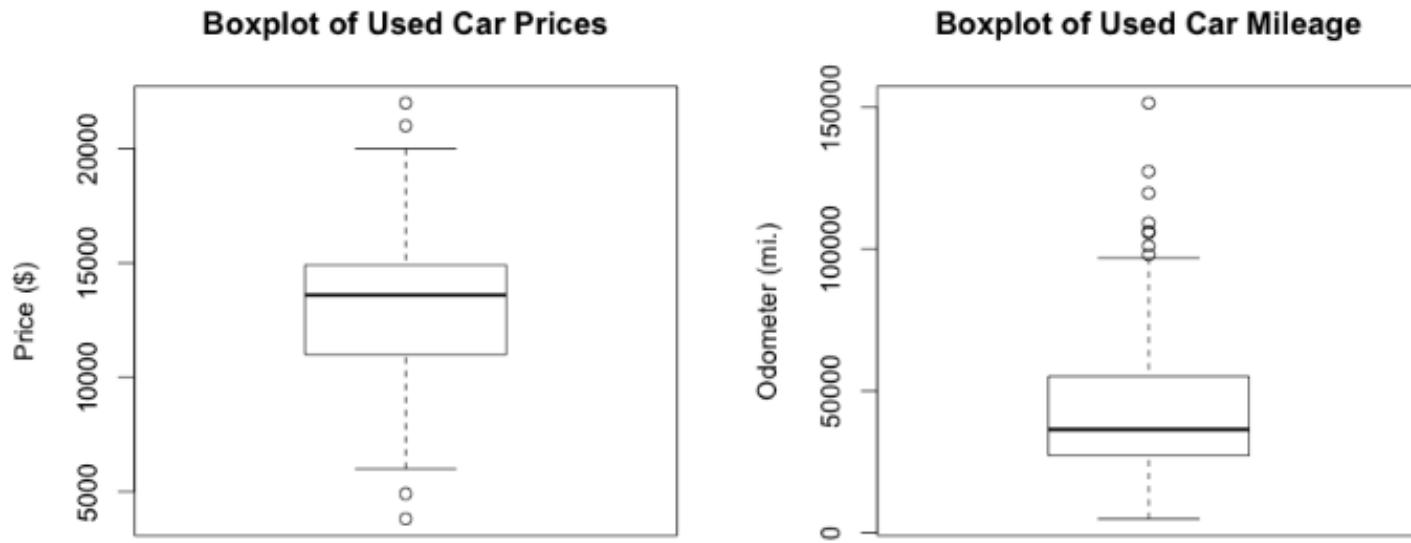
```
##pengisian dengan regresi (regression imputation)
reg <- lm(Lebar~Tinggi,data=datalengkap)
prediksi.lebar <- predict(reg,newdata=datahilang)
prediksi.lebar
Lebar.3 <- dataikan$Lebar
Lebar.3[c(4,23)] <- prediksi.lebar
datalengkap.3 <- data.frame(dataikan[,1:2],Lebar.3)
head(datalengkap.3)
model.1 <- lm(Tinggi~.,data=datalengkap.3)
model.1
model.2 <- lm(Tinggi~Panjang,data=datalengkap.3)
model.2
```

Model 1 dan model 2 menggunakan data lengkap dengan pengamatan ke-4 dan ke-23 diisi dengan nilai dugaan regresi

Pencilan (Outlier)

- Pencilan adalah titik dimana nilainya jauh dari nilai secara umum (yang diprediksi oleh model).
- Pencilan dapat muncul karena berbagai alasan, seperti pencatatan observasi yang salah selama pengumpulan data.

Ilustrasi pencilan dalam kasus univariat



- Nilai minimum dan maksimum diilustrasikan menggunakan garis yang memanjang di bawah dan di atas kotak; namun, sudah menjadi konvensi untuk hanya mengizinkan garis diperpanjang hingga minimum atau maksimum 1.5 kali IQR di bawah Q1 atau di atas Q3. Setiap nilai yang berada di luar ambang ini dianggap sebagai pencilan dan dilambangkan sebagai lingkaran atau titik.
- Misalnya, ingatlah bahwa IQR untuk variabel harga (Prices) adalah 3909 dengan $Q1=10995$ dan $Q3=14904$. Oleh karena itu, pencilan adalah nilai yang kurang dari $10995 - 1.5 * 3905 = 5137.5$ atau lebih besar dari $14904 + 1.5 * 3905 = 20761.5$.

Ilustrasi pencilan dalam kasus regresi linier

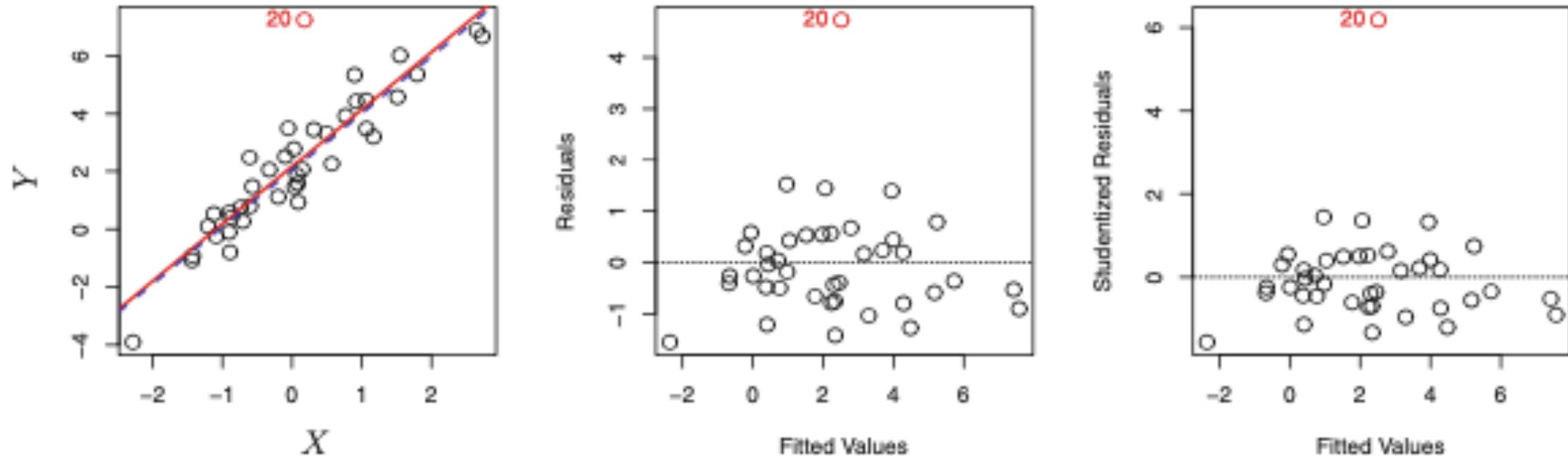


FIGURE 3.12. Left: The least squares regression line is shown in red, and the regression line after removing the outlier is shown in blue. Center: The residual plot clearly identifies the outlier. Right: The outlier has a studentized residual of 6; typically we expect values between -3 and 3 .

- Dalam hal ini, menghapus pencilan memiliki efek yang kecil pada garis MKT: hal itu menyebabkan hampir tidak ada perubahan pada kemiringan, dan pengurangan intersep yang sangat kecil.
- Hal tersebut adalah tipikal untuk pencilan yang tidak memiliki nilai prediktor yang tidak biasa memiliki sedikit efek pada kuadrat terkecil.
- Namun, bahkan jika pencilan tidak memiliki banyak pengaruh pada dugaan MKT, hal itu dapat menyebabkan masalah lain.
 - Nilai JKG berubah → perubahan selang kepercayaan, p-value, R^2
- Pengamatan yang mutlak studentized residualnya lebih besar dari 3 dapat diidentifikasi sebagai pencilan.

- Studentized Residual

Misalkan diketahui model regresi linier sebagai berikut:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

dengan

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix}$$

maka hat-matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$, dengan diagonal ke- i matriks \mathbf{H} adalah h_{ii} .

Akibatnya, studentized residual dapat dicari dengan:

$$t_i = \frac{\hat{\varepsilon}_i}{\hat{\sigma} \sqrt{1 - h_{ii}}}$$

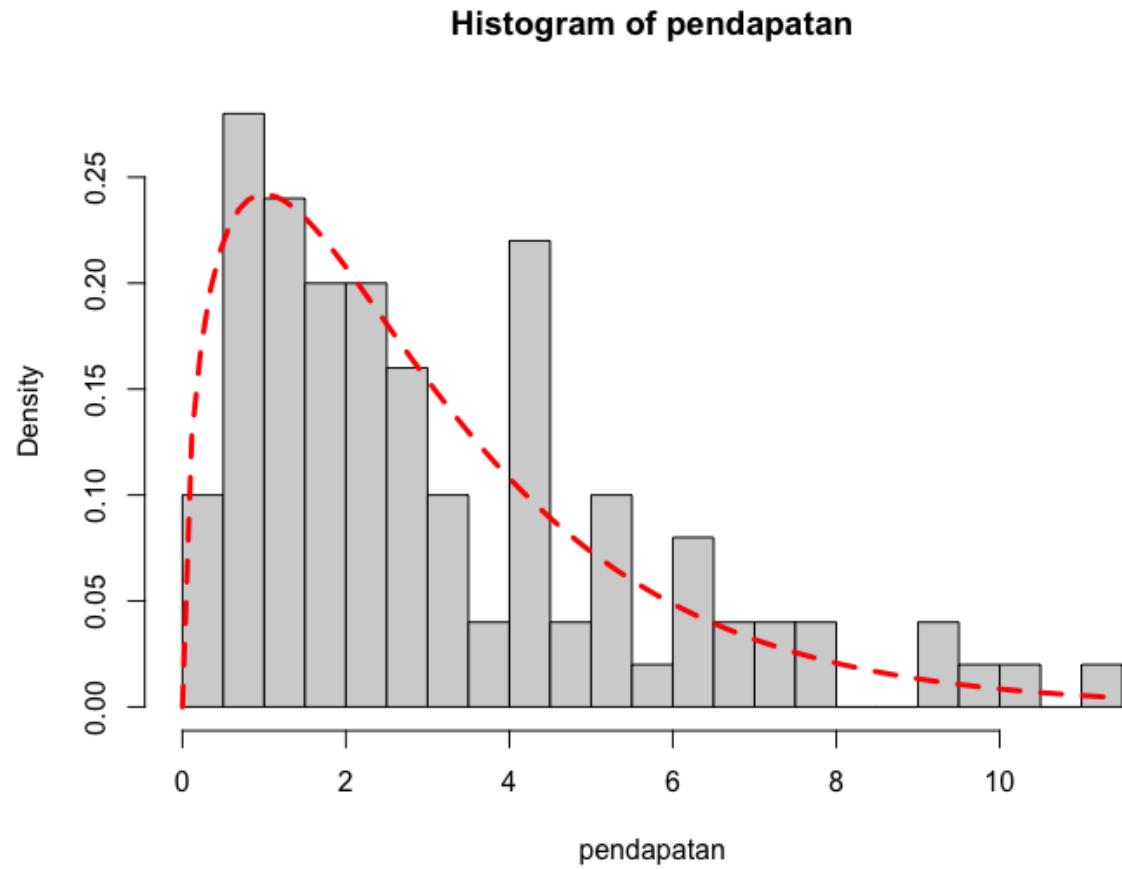
dengan $\hat{\sigma}^2 = \frac{1}{n-m} \sum_{j=1}^n \hat{\varepsilon}_j^2$; m banyaknya parameter

- Jika kita yakin bahwa pencilan telah terjadi karena kesalahan dalam pengumpulan atau pencatatan data, maka salah satu solusinya adalah dengan menghilangkan observasi tersebut.
- Namun, kehati-hatian harus dilakukan, karena pencilan malah dapat menunjukkan kekurangan model, seperti prediktor yang hilang.

Transformasi Data

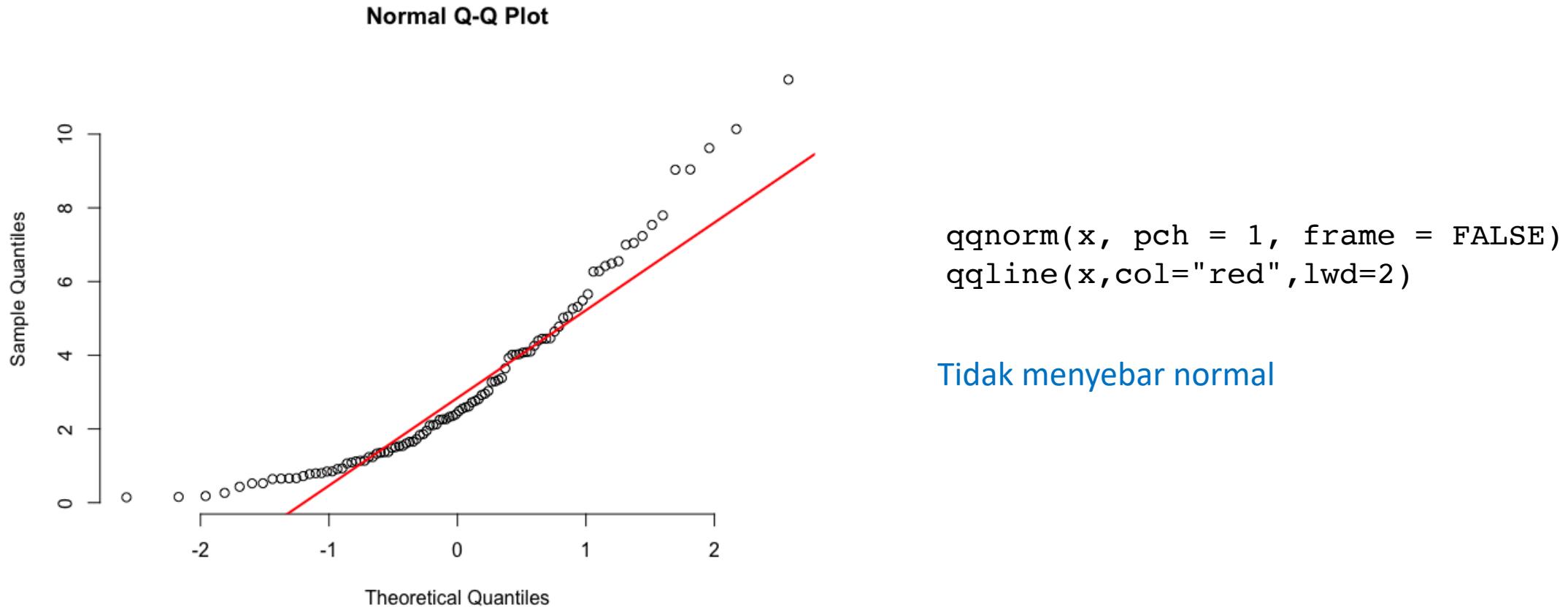
- Pada beberapa kasus, seringkali ditemui sebaran peubah yang diteliti tidak sesuai dengan asumsi.
- Misalkan pada kasus peubah univariat yang mengharuskan memiliki sebaran normal yang simetris, seringkali ditemui bahwa nilai penyebarannya tidak simetris.
 - Contoh: Peubah pendapatan, ingin dilakukan uji hipotesis yang mengharuskan memiliki sebaran normal

Ilustrasi



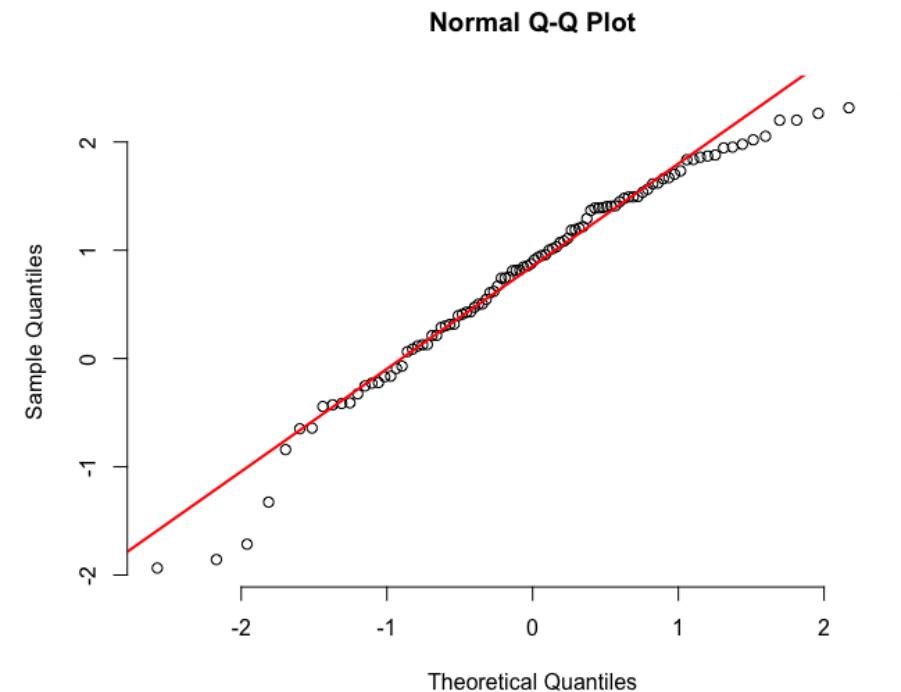
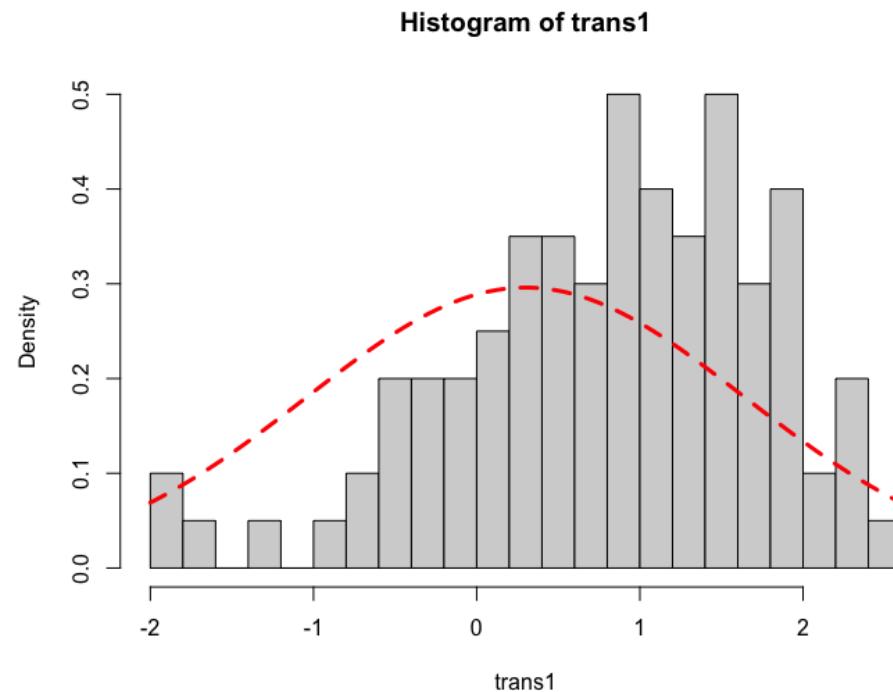
```
set.seed(123456)
pendapatan <- rchisq(100,3)
hist(pendapatan,breaks=30,freq=F)
x <- pendapatan
curve(dchisq(x,3),lty=2,col="red",lwd=3,add=T)
```

Tidak simetri (miring ke kanan)



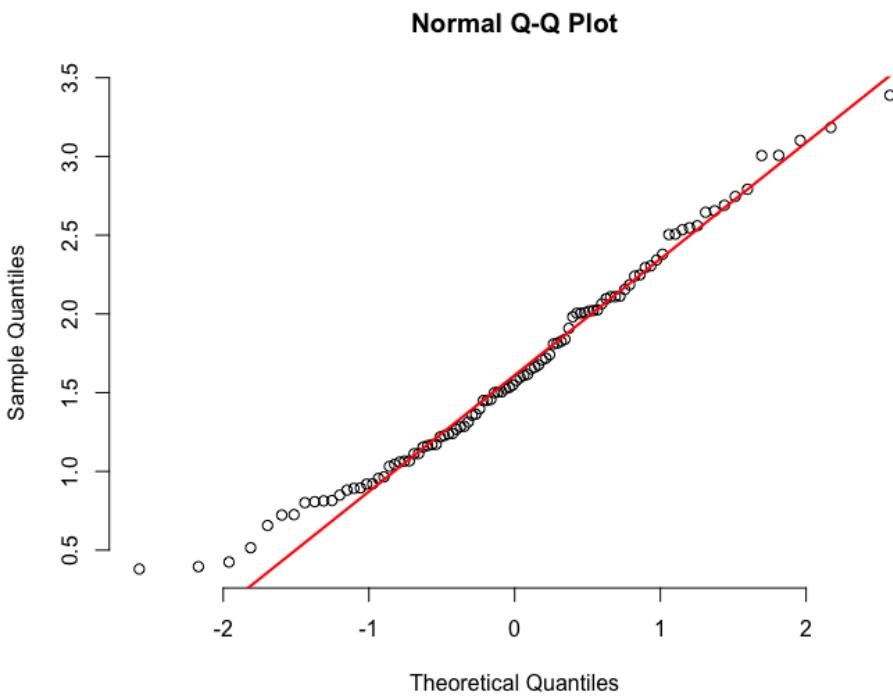
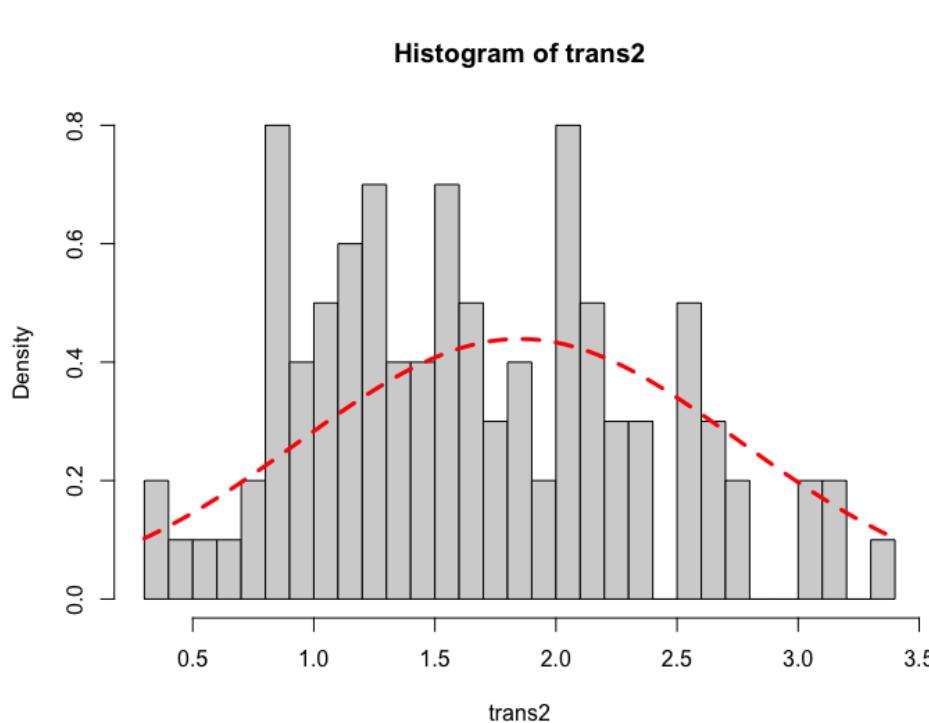
Transformasi → dapat dicobakan transformasi $\ln(Y)$ atau \sqrt{Y}

transformasi $\ln(Y)$



```
trans1 <- log(x)
hist(trans1, breaks=30, freq=F)
x <- trans1
curve(dnorm(x, mean(x), sd(x)), lty=2, col="red", lwd=3, add=T)
qqnorm(x, pch = 1, frame = FALSE)
qqline(x, col="red", lwd=2)
```

transformasi \sqrt{Y}



```
trans2 <- sqrt(pendapatan)
hist(trans2, breaks=30, freq=F)
x <- trans2
curve(dnorm(x, mean(x), sd(x)), lty=2, col="red", lwd=3, add=T)
qqnorm(x, pch = 1, frame = FALSE)
qqline(x, col="red", lwd=2)
```

Metode Box-Cox

- Transformasi Box Cox adalah transformasi peubah (peubah Y) yang tidak normal menjadi bentuk normal.
- Normalitas adalah asumsi penting untuk banyak metode statistik; jika data tidak normal, dengan menerapkan Box-Cox dapat menjalankan lebih banyak pengujian.
- Transformasi Box Cox dinamai ahli statistik George Box dan Sir David Roxbee Cox yang berkolaborasi pada makalah tahun 1964 dan mengembangkan teknik tersebut.

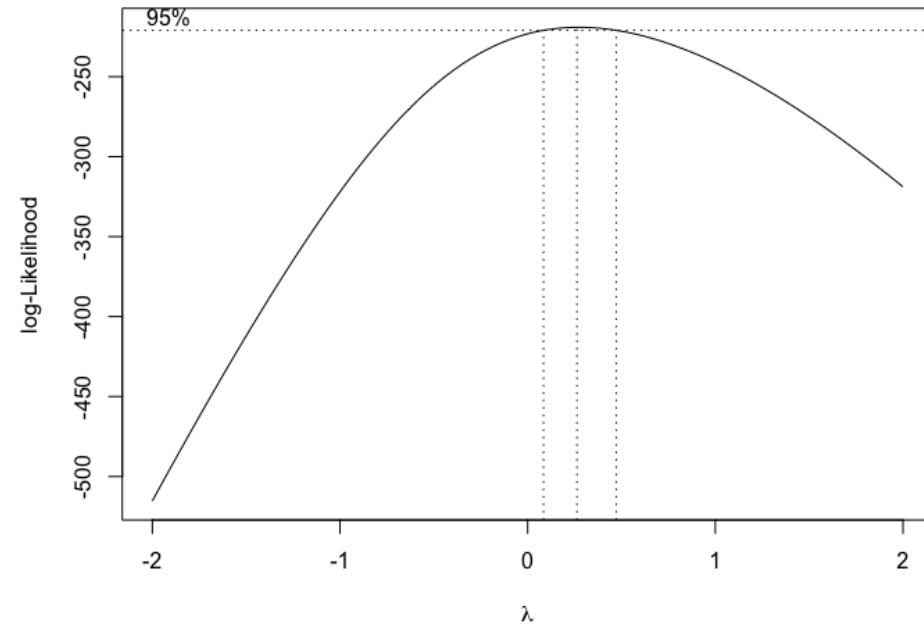
- Inti dari transformasi Box Cox adalah pangkat lambda (λ), yang bervariasi dari -5 hingga 5.
- Semua nilai λ dipertimbangkan dan nilai optimal untuk data yang akan dipilih; "Nilai optimal" adalah yang menghasilkan dugaan terbaik dari kurva sebaran normal.
- Transformasi Y memiliki bentuk:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & ; \lambda \neq 0 \\ \ln(y) & ; \lambda = 0 \end{cases}$$

- Namun, transformasi yang paling umum dijelaskan dalam tabel berikut:

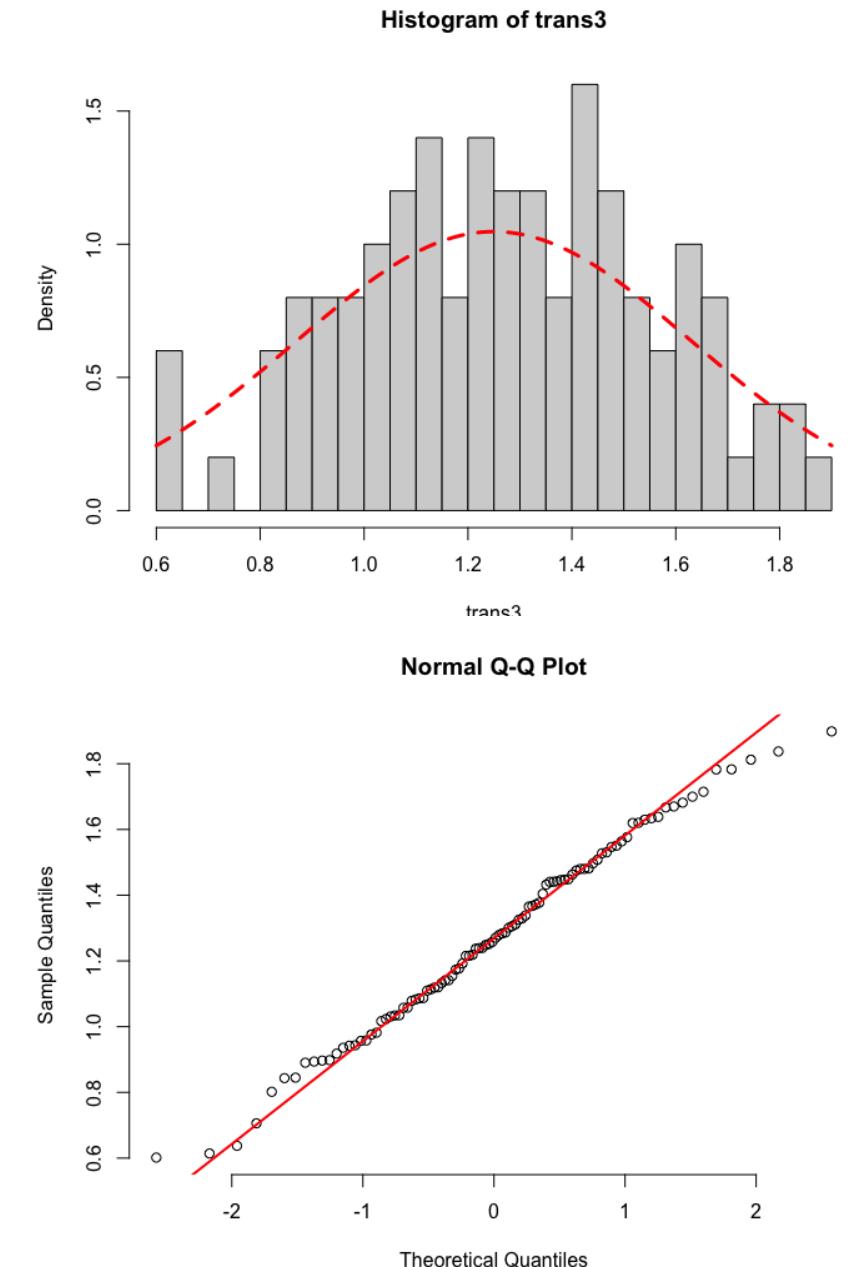
Nilai λ	Transformasi Y
-3	$Y^{-3} = 1/Y^3$
-2	$Y^{-2} = 1/Y^2$
-1	$Y^{-1} = 1/Y$
-0.5	$Y^{-0.5} = 1/\sqrt{Y}$
0	$\ln(Y)$
0.5	$Y^{0.5} = \sqrt{Y}$
1	$Y^1 = Y$
2	Y^2
3	Y^3

Transformasi boxcox

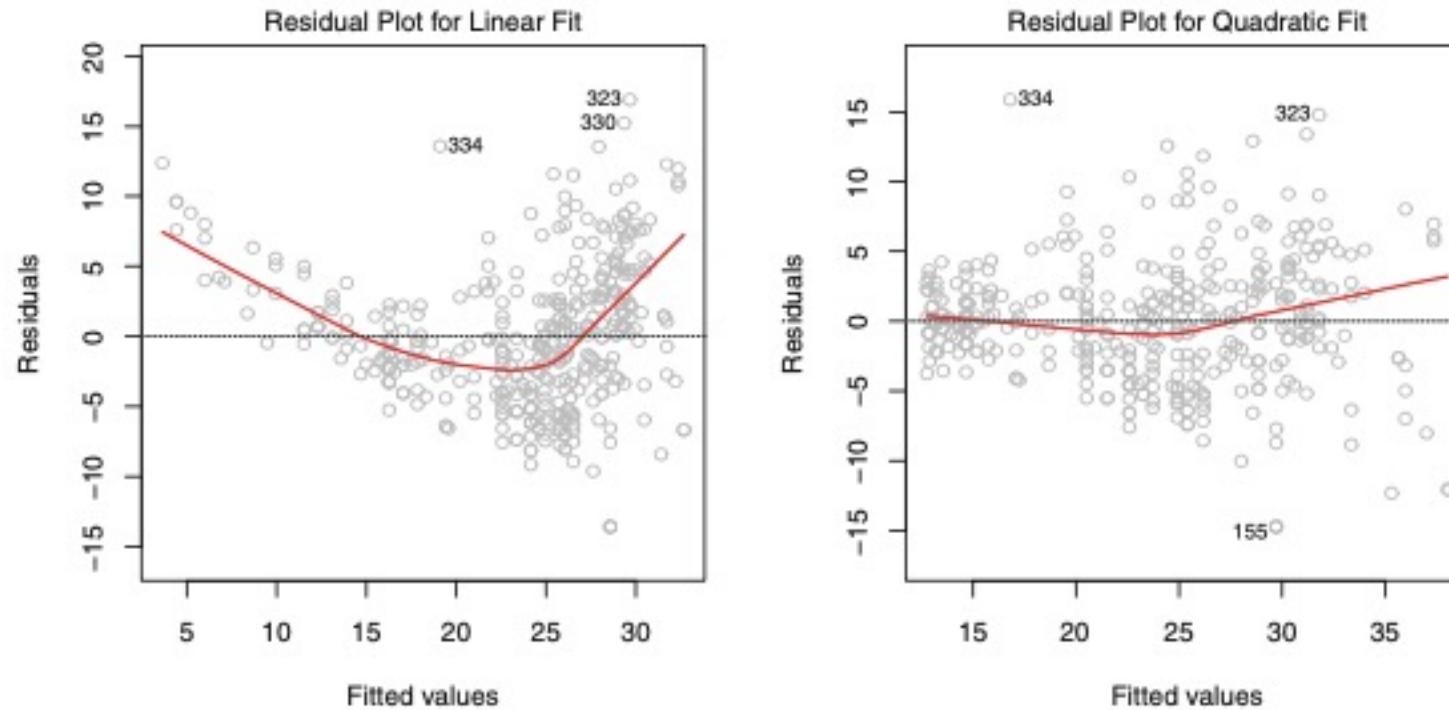


```
library(MASS)
bb <- boxcox(lm(pendapatan~1))
lambda <- bb$x[which.max(bb$y)]
lambda

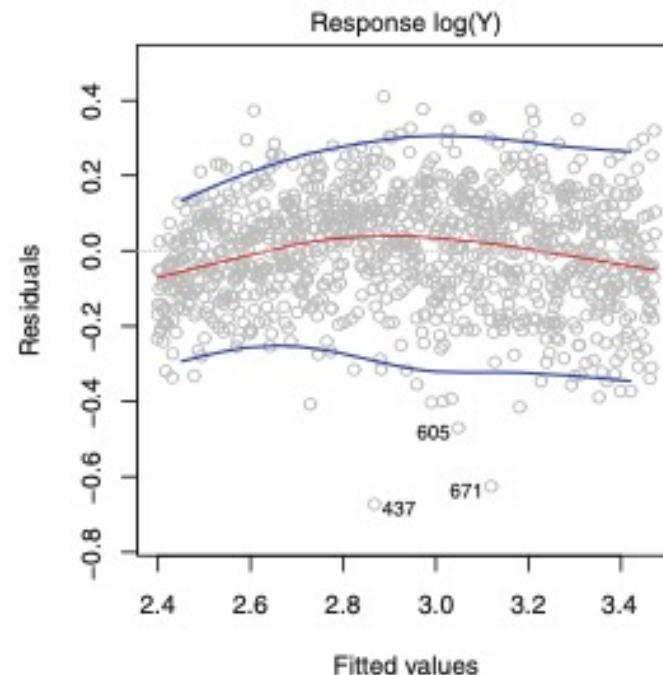
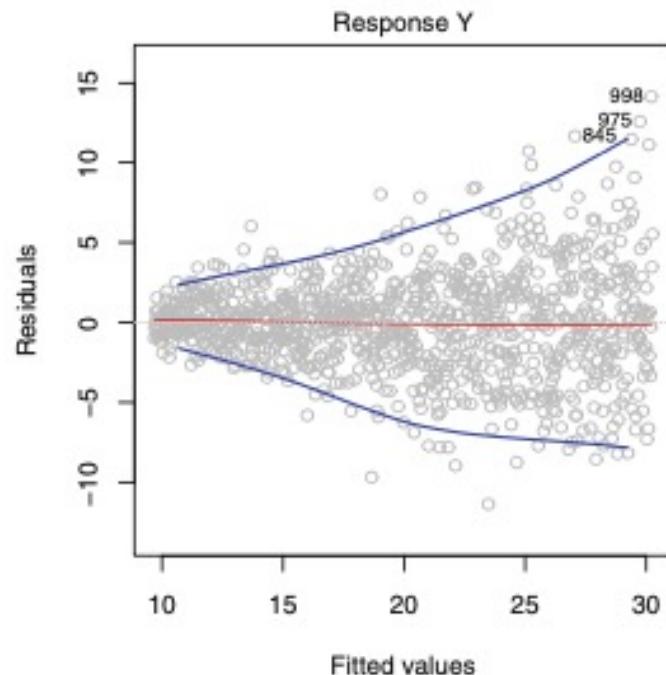
trans3 <- pendapatan^(lambda)
hist(trans3,breaks=30,freq=F)
x <- trans3
curve(dnorm(x,mean(x),sd(x)),lty=2,col="red",lwd=3,add=T)
qqnorm(x, pch = 1, frame = FALSE)
qqline(x,col="red",lwd=2)
```



- Dalam kasus regresi, jika plot residual menunjukkan bahwa ada hubungan non-linear dalam data, maka pendekatan sederhana adalah dengan menggunakan transformasi prediktor non-linear, seperti $\ln(X)$, \sqrt{X} , dan X^2 , dalam model regresi. Pendekatan non-linier ini dapat berupa pemodelan regresi non-linier.



- Ketika dihadapkan dengan masalah ketidakhomogenan ragam residual (heteroskedastisitas dalam kasus regresi), salah satu solusi yang mungkin adalah dengan transformasi Y menggunakan fungsi cekung seperti $\ln(Y)$ atau \sqrt{Y} .
- Transformasi seperti itu menghasilkan jumlah penyusutan yang lebih besar dari respons yang lebih besar, yang mengarah ke pengurangan heteroskedastisitas.



Latihan (isian singkat):

1. Seseorang tidak menghadiri tes narkoba karena orang tersebut menggunakan narkoba pada malam sebelumnya, merupakan contoh mekanisme data hilang ...
2. Hilangnya pengamatan pada peubah jarak tempuh mobil pada pengumpulan data mengenai mobil bekas, yang notabenenya nilai peubah ini dapat diprediksi oleh peubah tahun mobil diproduksi, merupakan contoh mekanisme data hilang ...
3. Tidak terisinya pengamatan pada variabel IPK karena terdapat kerusakan pada sistem entri data, merupakan contoh mekanisme data hilang ...
4. Metode penanganan data hilang yang cenderung bersifat underestimate adalah ...
5. Asumsi mekanisme data hilang yang harus dipenuhi agar metode pengisian dengan regresi valid adalah data hilang mengikuti mekanisme ...

6. Metode penanganan data hilang yang valid diterapkan apabila mekanismenya MCAR dan ukuran contoh yang cukup banyak adalah ...
7. Kriteria pengamatan pencilan pada boxplot adalah ...
8. Berdasarkan studentized residual, pengamatan dalam regresi yang dianggap sebagai pencilan adalah ...
9. Untuk nilai $\lambda=0$ pada transformasi box cox, maka transformasi yang dilakukan pada peubah Y adalah ...
10. Transformasi pada peubah prediktor (X) pada model regresi dimaksudkan untuk ...

- Silahkan enroll di newlms MK STA1382 Teknik Pembelajaran Mesin dengan enrollment key:

sta1382ok

Terima kasih 😊



Analisis Regresi Beserta Metode Evaluasinya

Kuliah 3 - STA1382 Teknik
Pembelajaran Mesin

Septian Rahardiantoro



Outline

- Pengantar Pemodelan Statistika
- Regresi linier beserta metode evaluasinya
- Regresi logistik beserta metode evaluasinya

Pengantar Pemodelan Statistika

$$y = f(x_1, x_2, \dots, x_p) + \varepsilon$$

- Membangun miniatur dari dunia nyata
 - dinyatakan dalam satu atau beberapa fungsi matematis
- Menyederhanakan fenomena nyata sehingga mudah memahami pola umum yang ada
 - memberikan penjelasan terhadap perubahan
 - memberikan penjelasan tentang perbedaan yang terjadi
 - menemukan faktor yang menyebabkan perubahan dan perbedaan

Pemodelan

- Tujuan/Manfaat:
 - Sering digunakan untuk meng-explore dataset yang dimiliki
 - Digunakan untuk melakukan prediksi berdasarkan informasi dari variabel prediktor
 - Digunakan untuk mengkaji dan memahami bagaimana suatu variabel berhubungan dengan variabel yang lain
- Are not perfect
 - “All models are wrong, but some are useful” (GEP Box)

Beberapa Model Statistika yang Populer

Jenis Variabel Target	Model Statistika
Numerik	Regresi Linier
Kategorik	Regresi Logistik Pohon Klasifikasi (Classification Tree)

Regresi Linier

- Syarat Utama: Variabel output (Y) bersifat numerik
- Variabel prediktor (X)
 - numerik OK, kategorik OK
 - satu OK, lebih dari satu OK
- Bentuk model

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$$

- **Analisis Regresi** digunakan untuk:
 - Menjelaskan dampak perubahan peubah prediktor terhadap peubah respon
 - Memprediksi nilai dari peubah respon berdasarkan nilai dari setidaknya sebuah peubah prediktor

Peubah Respon (peubah tak bebas, peubah terikat, dependent variable):
peubah yang ingin kita jelaskan

Peubah Prediktor (peubah bebas, independent variable): peubah yang
digunakan untuk menjelaskan peubah respon

Regresi Linier Sederhana

- Suatu pendekatan untuk memprediksi peubah respon kuantitatif Y berdasarkan sebuah peubah prediktor X
- Pendekatan ini mengasumsikan bahwa ada hubungan linier antara X dan Y

The population regression model:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Diagram illustrating the components of the population regression model:

- Dependent Variable: y
- Population y intercept: β_0
- Population Slope Coefficient: β_1
- Independent Variable: x
- Random Error term, or residual: ϵ

The equation is divided into two main components:

- Linear component: $\beta_0 + \beta_1 x$
- Random Error component: ϵ

- Pendugaan koefisien
 - Misalkan $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ adalah prediksi untuk Y berdasarkan nilai ke- i peubah X (dengan $i = 1, 2, 3, \dots, n$)
 - Maka residual ke- i didefinisikan oleh:

$$e_i = y_i - \hat{y}_i \rightarrow e_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i$$
 - JKG (Jumlah Kuadrat Galat) didefinisikan oleh:

$$JKG = e_1^2 + e_2^2 + \cdots + e_n^2$$

$$JKG = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$
 - Penduga MKT (Metode Kuadrat Terkecil), memilih $\hat{\beta}_0$ dan $\hat{\beta}_1$ yang meminimumkan JKG . Dengan perhitungan kalkulus diperoleh:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}; \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Ilustrasi kontur dan plot 3D pada JKG (RSS) untuk model dengan $Y = \text{sales}$ dan $X = \text{TV}$

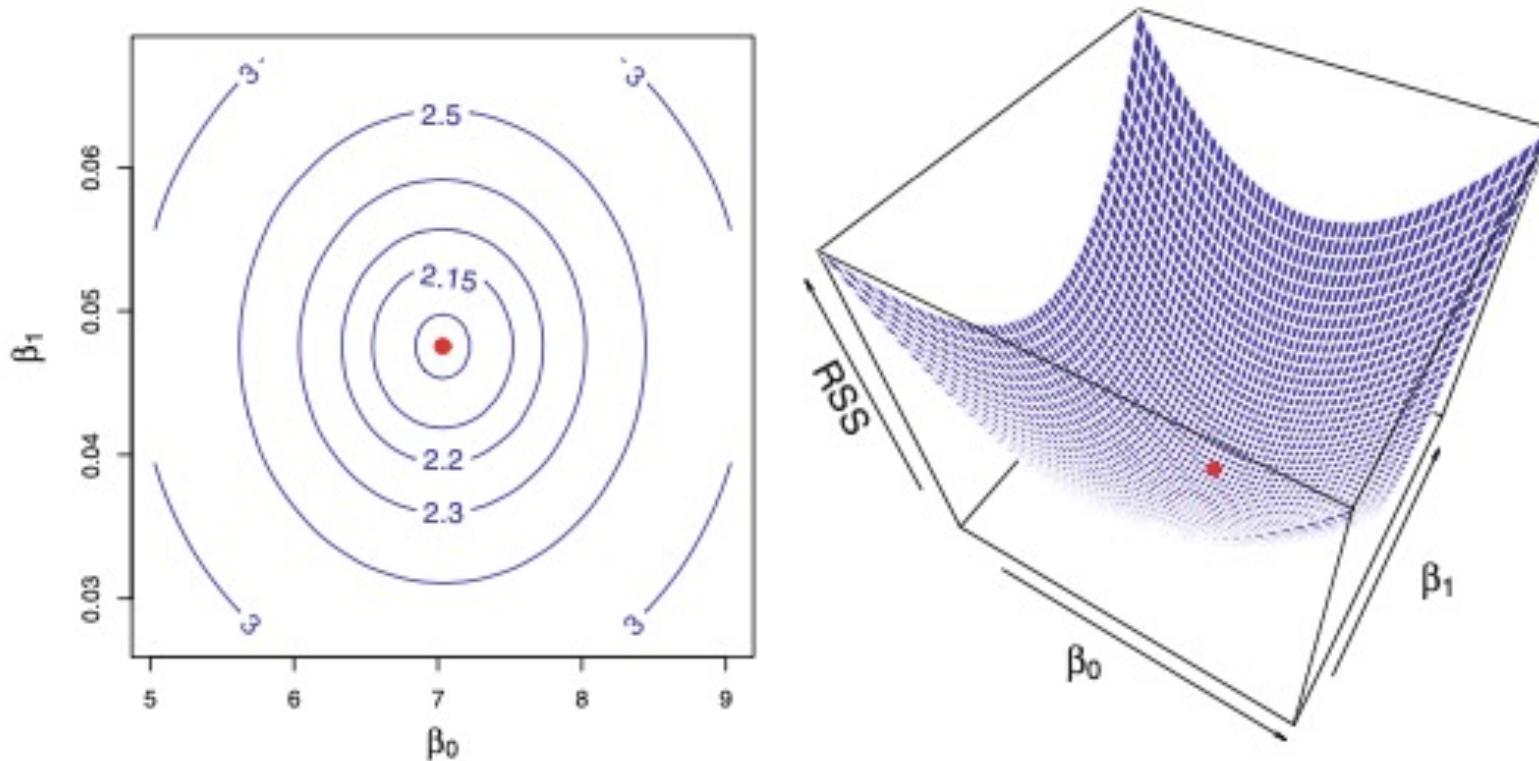
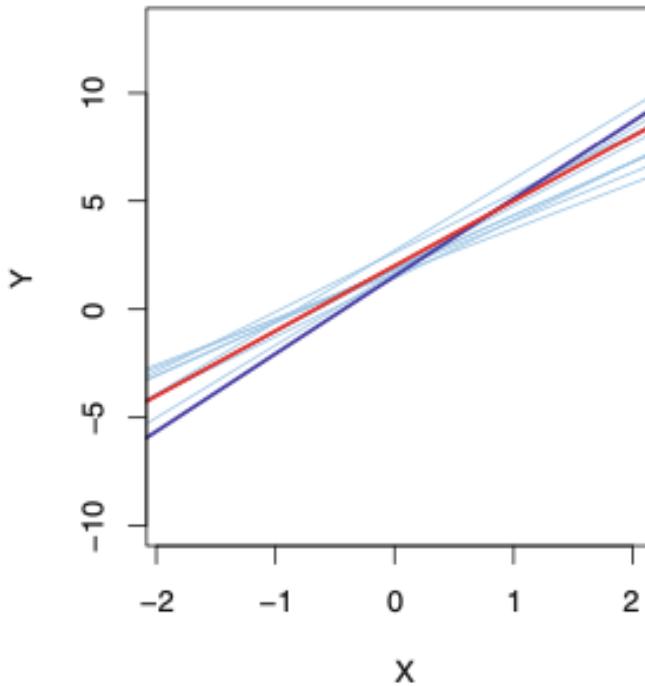
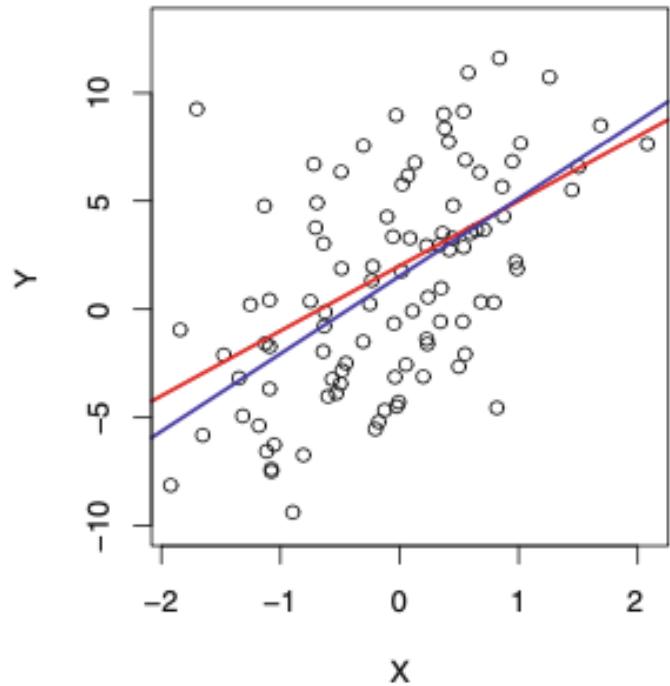


FIGURE 3.2. Contour and three-dimensional plots of the RSS on the Advertising data, using sales as the response and TV as the predictor. The red dots correspond to the least squares estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, given by (3.4).

Menilai Akurasi Penduga Koefisien



Simulasi.

Kiri: Garis merah mewakili hubungan sebenarnya, $f(X) = 2 + 3X$, yang dikenal sebagai garis regresi populasi. Garis biru adalah garis kuadrat terkecil (MKT); yang merupakan dugaan kuadrat terkecil untuk $f(X)$ berdasarkan data yang diamati, ditampilkan dalam warna hitam.

Kanan: Garis regresi populasi ditampilkan lagi dengan warna merah, dan garis kuadrat terkecil berwarna biru tua. Dengan warna biru muda, sepuluh garis kuadrat terkecil ditampilkan, masing-masing dihitung berdasarkan kumpulan pengamatan acak yang terpisah. Setiap garis kuadrat terkecil berbeda, tetapi rata-rata garis kuadrat cukup dekat dengan garis regresi populasi.

- Sekilas, perbedaan antara garis regresi populasi dan garis kuadrat terkecil mungkin tampak halus dan membingungkan.
- Dalam kasus ini, diketahui satu kumpulan data, namun terdapat banyak garis berbeda menggambarkan hubungan antara prediktor dan respons?
- Sehingga, muncul pertanyaan seberapa dekat penduga $\hat{\beta}_0$ dan $\hat{\beta}_1$ terhadap β_0 dan β_1
 - Hal ini dapat diselidiki dengan standar error (galat baku) $\hat{\beta}_0$ dan $\hat{\beta}_1$

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right]; SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

- Secara kasar, galat baku memberi tahu kita jumlah rata-rata perkiraan pendugaan berbeda dari nilai parameter aktualnya
- Selang kepercayaan bagi β_0 dan β_1 (taraf nyata 95%)

$$\hat{\beta}_0 \pm 2 \times SE(\hat{\beta}_0); \quad \hat{\beta}_1 \pm 2 \times SE(\hat{\beta}_1)$$

- Uji hipotesis $\beta_1 \rightarrow H_0: \beta_1 = 0; H_1: \beta_1 \neq 0$ dengan statistik uji $t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$

Menilai Akurasi Model

- Kualitas kecocokan regresi linier biasanya dinilai menggunakan dua besaran terkait: Galat Baku Residual (residual standard error) dan statistik R^2

- **Galat Baku Residual**

- Galat Baku Residual merupakan dugaan simpangan baku dari residual, yakni jumlah rata-rata respon yang akan menyimpang dari garis regresi yang sebenarnya.

$$\text{Galat Baku Residual} = \sqrt{\frac{1}{n-2} JKG} = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- Galat Baku Residual dianggap sebagai ukuran kecocokan model dengan data.
 - Jika prediksi yang diperoleh dengan menggunakan model sangat dekat dengan nilai hasil sebenarnya—yaitu, jika $\hat{y}_i \approx y_i$ untuk $i = 1, \dots, n$ —maka Galat Baku Residual akan menjadi kecil, dan kita dapat menyimpulkan bahwa model tersebut sangat cocok dengan data.
 - Di sisi lain, jika \hat{y}_i sangat jauh dari y_i untuk satu atau lebih pengamatan, maka Galat Baku Residual mungkin cukup besar, menunjukkan bahwa model tidak sesuai dengan data dengan baik.

- **Statistik R^2**

- Galat Baku Residual memberikan ukuran mutlak ketidaksesuaian model dengan data.
- Tetapi karena diukur dalam satuan Y , tidak selalu jelas apa yang dimaksud dengan Galat Baku Residual yang baik.
- Statistik R^2 memberikan alternatif ukuran kecocokan model.
- Bentuknya berupa proporsi (proporsi ragam yang dijelaskan) sehingga selalu mengambil nilai antara 0 dan 1, dan tidak bergantung pada skala Y .

$$R^2 = \frac{JKT - JKG}{JKT} = 1 - \frac{JKG}{JKT} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}$$

- R^2 mengukur proporsi keragaman dalam Y yang dapat dijelaskan dengan menggunakan X .

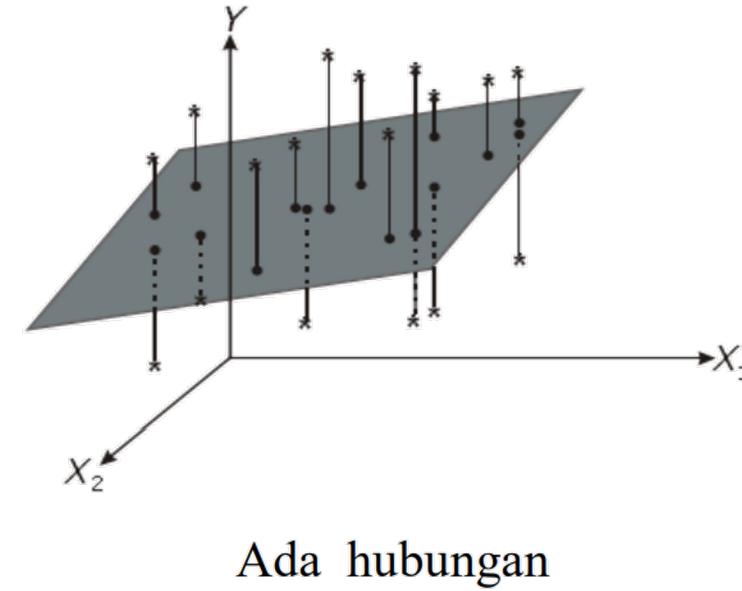
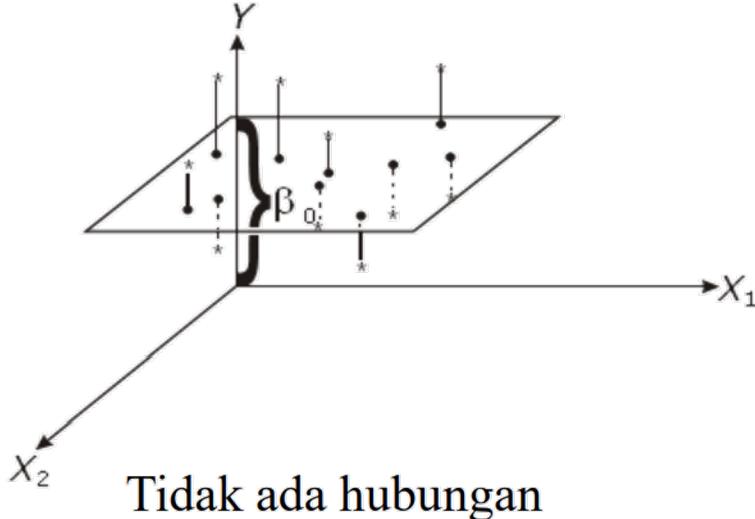
Regresi Linier Berganda

- Analisis regresi linear berganda:

- Secara umum, kita memodelkan peubah respon Y sebagai fungsi linier dari k peubah prediktor (X) sebagai:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \varepsilon$$

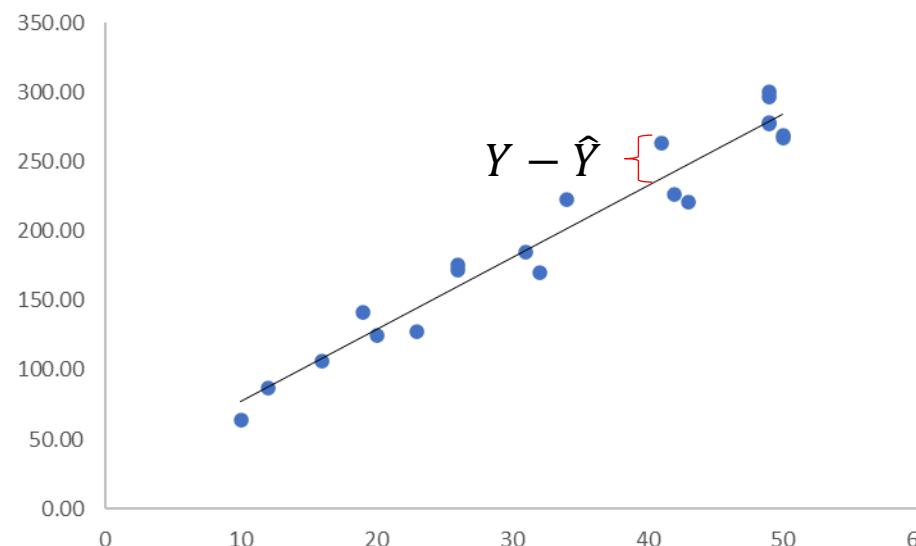
- Atau dalam notasi matriks $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$
- Jika kita memiliki dua variabel X , model dapat diilustrasikan sebagai berikut



- Pendugaan koefisien regresi:

- Pendugaan koefisien regresi diperoleh dengan meminimumkan jumlah kuadrat galat (residual) → OLS (Ordinary Least Square) atau MKT (Metode Kuadrat Terkecil)
- Dalam hal ini dicari dugaan dari $\beta_j, j = 0, 1, 2 \dots, k$ yang meminimumkan $\sum_i \varepsilon^2$, dengan $\varepsilon = Y - \hat{Y}$, yang dalam notasi matriks diperoleh

$$\hat{\beta} = (X'X)^{-1}X'y$$



Asumsi model regresi linear

Nilai mean dari peubah Y dimodelkan secara akurat oleh fungsi linier dari peubah-peubah X

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \varepsilon$$

Antar peubah X tidak ada multikolinearitas

Galat acak diasumsikan menyebar normal dengan nilai tengah nol dan memiliki ragam yang konstan σ^2 (ragam homogen)

Galat bersifat independen/saling bebas (tidak ada autokorelasi)

Menilai Akurasi Model

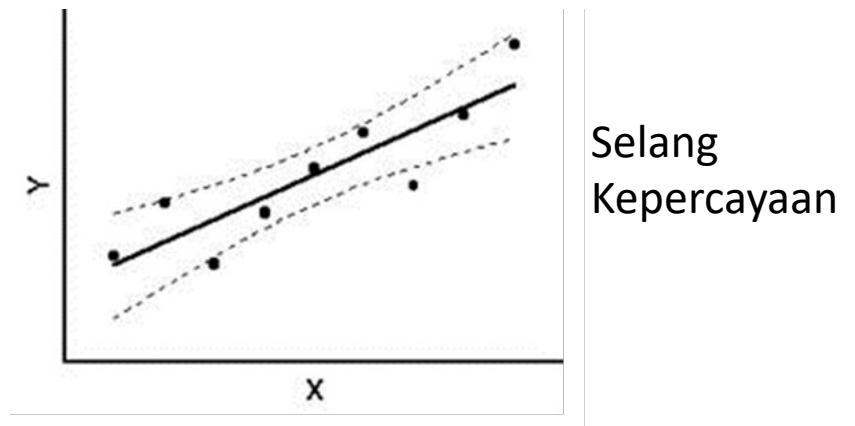
- Dua ukuran numerik yang paling umum untuk mengidentifikasi kecocokan model adalah Galat Baku Residual dan R^2 .
- Nilai-nilai ini dihitung dan ditafsirkan dengan cara yang sama seperti untuk regresi linier sederhana.

$$\text{Galat Baku Residual} = \sqrt{\frac{1}{n-p-1} JKG} = \sqrt{\frac{1}{n-p-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Menilai Akurasi Prediksi

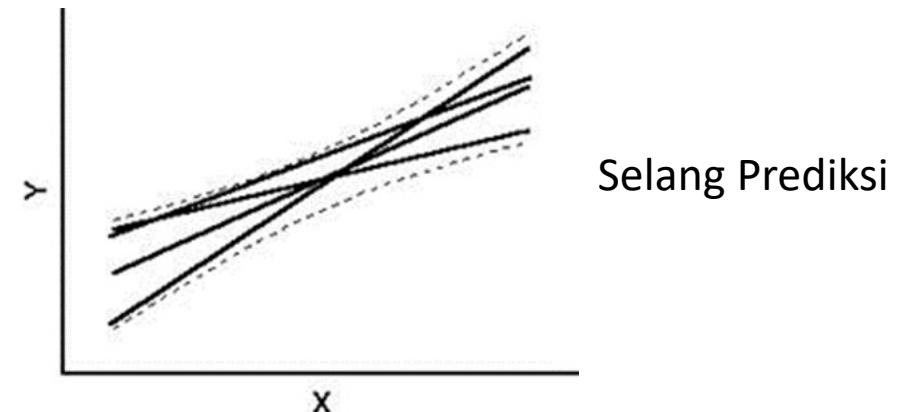
- Pemodelan prediktif merupakan masalah pengembangan model menggunakan data historis untuk membuat prediksi pada data baru yang belum dimiliki jawabannya.
- Berikut ini beberapa ukuran evaluasi dalam konteks prediksi untuk model regresi
 1. Selang Kepercayaan Prediksi (Confidence Interval)

$$\hat{y}_h \pm t_{\left(1 - \frac{\alpha}{2}; n-2\right)} \times \sqrt{KTG \left(\frac{1}{n} + \frac{(x_h - \bar{x})^2}{\sum(x_i - \bar{x})^2} \right)}$$



2. Selang Prediksi (Prediction Interval)

$$\hat{y}_h \pm t_{\left(1 - \frac{\alpha}{2}; n-2\right)} \times \sqrt{KTG \left(1 + \frac{1}{n} + \frac{(x_h - \bar{x})^2}{\sum(x_i - \bar{x})^2} \right)}$$



3. MSE (Mean Squared Error) atau KTG

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

4. RMSE (Root Mean Squared Error)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

5. MAE (Mean Absolute Error)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Contoh 1

```
#simulasi analisis regresi
beta <- c(3,5,7)
set.seed(123456)
Xa <- matrix(rnorm(200,5,1),100,2)
X <- cbind(1,Xa)
e <- rnorm(100,0,4)
y <- X%*%beta+e
data1 <-
data.frame(y=y,X1=Xa[,1],X2=Xa[,2])

##model regresi
mod1 <- lm(y~X1+X2,data=data1)
summary(mod1)
```

```
> summary(mod1)

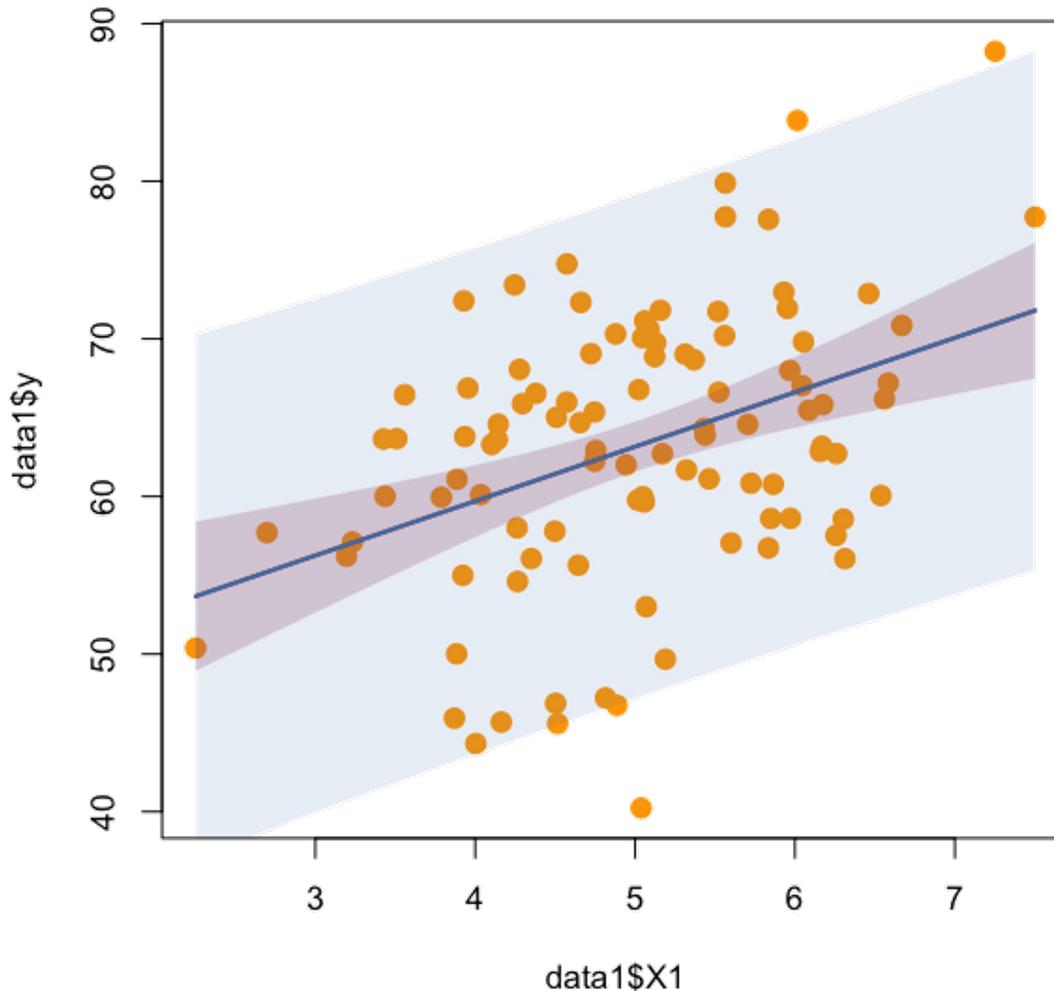
Call:
lm(formula = y ~ X1 + X2, data = data1)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.8883 -2.4990  0.0499  2.3688 10.2160 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.2994    3.2282   0.712   0.478    
X1          4.9915    0.4102  12.168  <2e-16 ***  
X2          7.2377    0.4142  17.473  <2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.961 on 97 degrees of freedom
Multiple R-squared:  0.7965,    Adjusted R-squared:  0.7923 
F-statistic: 189.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
plot(data1$X1,data1$y,pch=20,col="orange", cex=2)
library(DescTools)
mod2 <- lm(y~X1,data=data1)
lines(mod2,col="red") #conf interval
lines(mod2,col="steelblue", pred.level=0.95) #pred interval
```



```
##evaluasi  
yduga <- fitted.values(mod1)  
MSE <- mean((data1$y-yduga)^2)  
RMSE <- sqrt(MSE)  
MAE <- mean(abs(data1$y-yduga))
```

```
> MSE  
[1] 15.21544  
> RMSE  
[1] 3.900697  
> MAE  
[1] 3.029892
```

Regresi Logistik

- Model regresi yang diterapkan untuk peubah respon Y dengan skala kategorik
- Peubah respon Y dapat terdiri dari 2 kategori (biner), maupun lebih dari 2 kategori (multinomial) yang dapat urutan (ordinal) maupun tidak (nominal)
- Daripada memodelkan respon Y ini secara langsung, regresi logistik memodelkan peluang bahwa Y termasuk dalam kategori tertentu

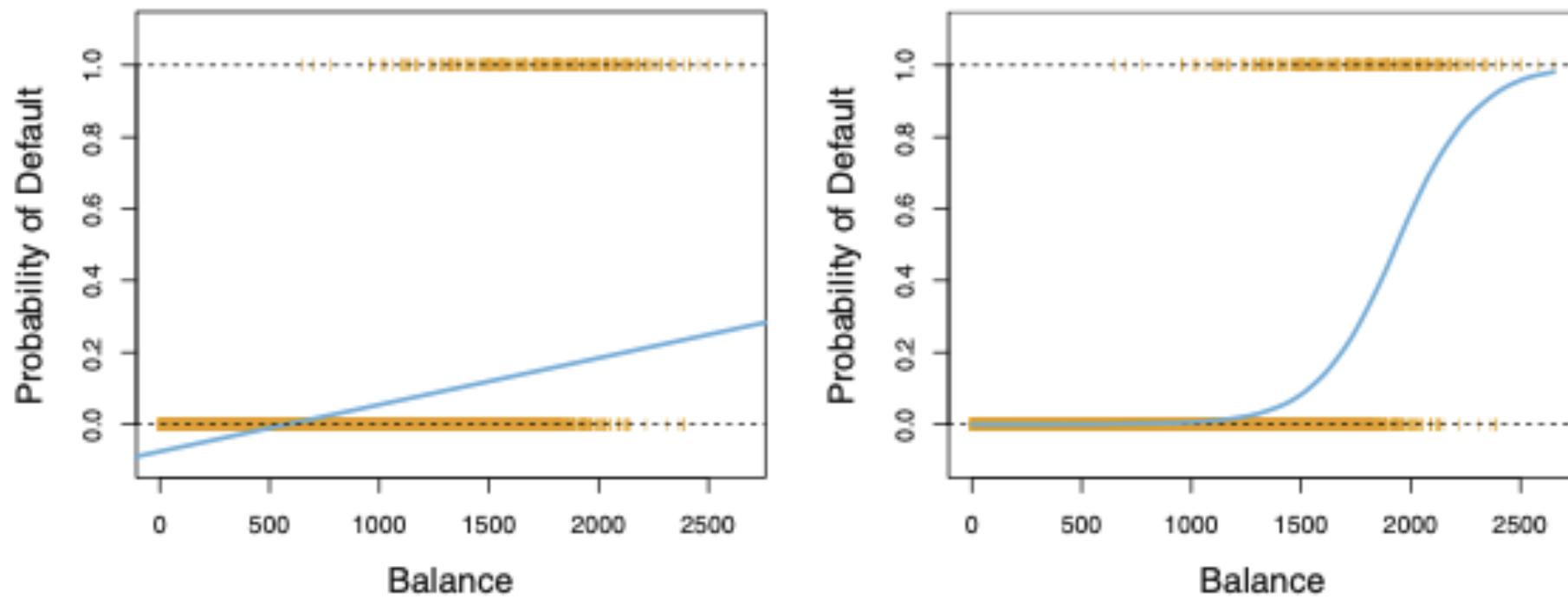


FIGURE 4.2. Classification using the `Default` data. Left: Estimated probability of `default` using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for `default`(No or Yes). Right: Predicted probabilities of `default` using logistic regression. All probabilities lie between 0 and 1.

Model Logistik

- Model hubungan antara $p(X) = P(Y = 1|X)$ dan X , dalam hal ini digunakan kode 0 atau 1 untuk kategori peubah respon

$$p(X) = \beta_0 + \beta_1 X$$

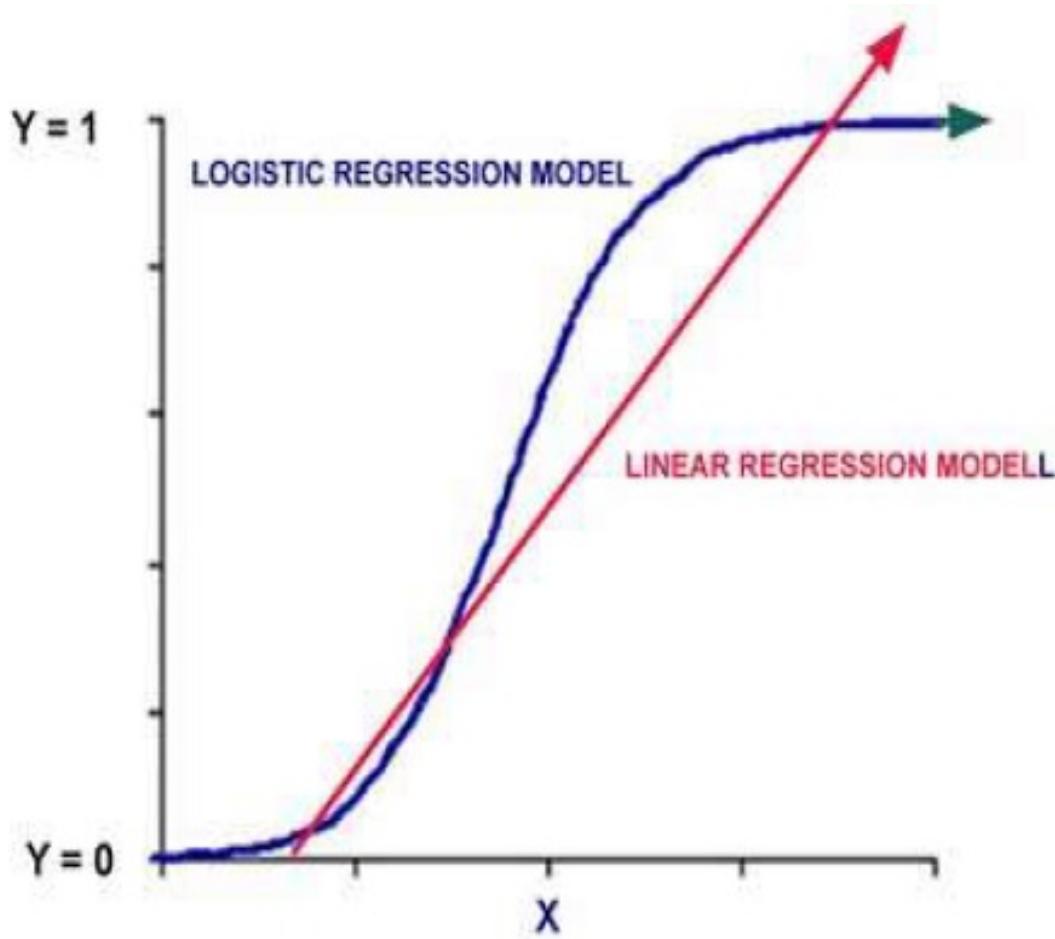
- Dengan fungsi logistik:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \rightarrow \frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

- Nilai $\frac{p(X)}{1-p(X)}$ disebut dengan odds, berkisar dari 0 s.d ∞
- Dengan menerapkan logaritma natural, maka diperoleh persamaan log-odds atau logit

$$\ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

- Dalam model regresi logistik, meningkatkan X sebesar satu unit mengubah log-odds sebesar β_1 , atau setara dengan mengalikan odds dengan e^{β_1}



- $\beta > 0$ maka kurva akan naik
- $\beta < 0$ maka kurva akan turun
- Jika $\beta = 0$ maka nilai berapapun nilai $p(X)$ konstan, berapapun nilai $X \rightarrow$ kurva akan menjadi garis horizontal

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Interpretasi nilai β_1

- 1 kenaikan X akan meningkatkan $\ln\left(\frac{p(X)}{1-p(X)}\right)$ sebesar β_1 satuan
- Dengan kata lain, 1 kenaikan X akan meningkatkan $\left(\frac{p(X)}{1-p(X)}\right)$ sebesar e^{β_1} satuan
- $\left(\frac{p(X)}{1-p(X)}\right)$ disebut dengan odds \rightarrow peluang dari kejadian terjadi dibagi dengan peluang dari kejadian tidak terjadi
- Artinya odds akan meningkat secara sebesar e^{β_1} untuk setiap kenaikan 1 unit X
- e^{β_1} : odds ratio (*OR*)

$$OR = e^{\beta_1} = \frac{odds(X = x + 1)}{odds(X = x)}$$

- An odds ratio indicates how much are likely, with respects to odds, a certain event occurs in one group relative to its occurrence in another group.

- Pendugaan koefisien regresi logistik
 - Menggunakan metode maksimum likelihood, dengan fungsi likelihood:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=1} (1 - p(x_{i'}))$$

- Sehingga, dugaan $\hat{\beta}_0$ dan $\hat{\beta}_1$ dipilih yang memaksimumkan nilai fungsi likelihood
- Prediksi
 - Peluang $p(X)$ dapat diprediksi dengan:
- Pada umumnya, jika $\hat{p}(X) \geq 0.5 \rightarrow \hat{y} = 1$ dan sebaliknya
- Model regresi logistik untuk lebih dari satu prediktor

$$\ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \rightarrow p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

Evaluasi Prediksi

Confusion Matrix

		Predicted Class			
		No	Yes	TN	True Negative
Observed Class	No	TN	FP	FP	False Positive
	Yes	FN	TP	FN	False Negative

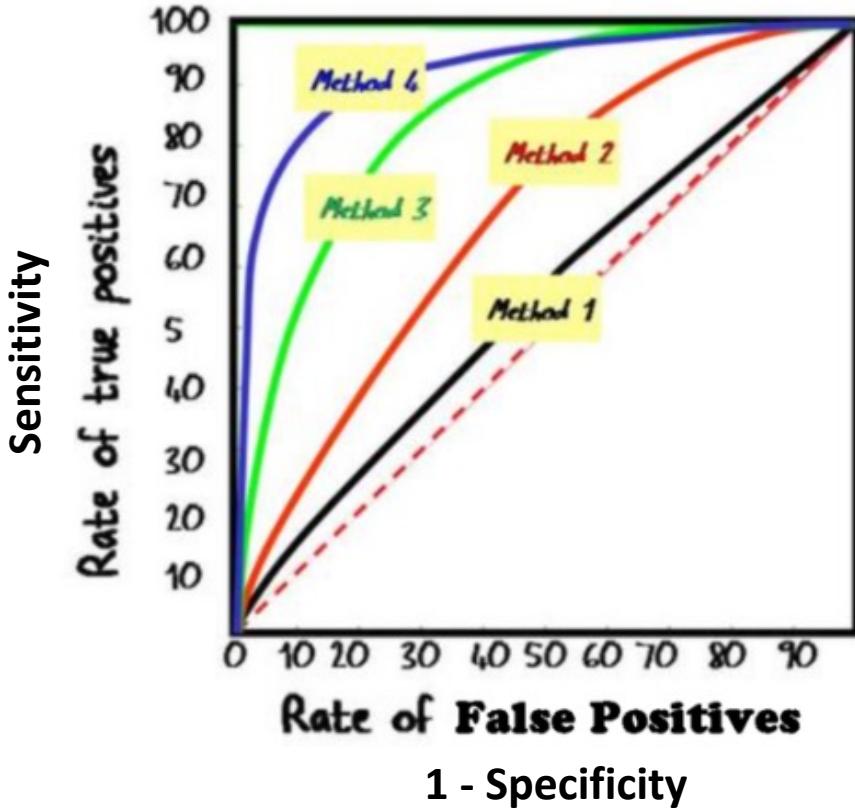
TP True Positive
TN True Negative
FP False Positive
FN False Negative

Model Performance

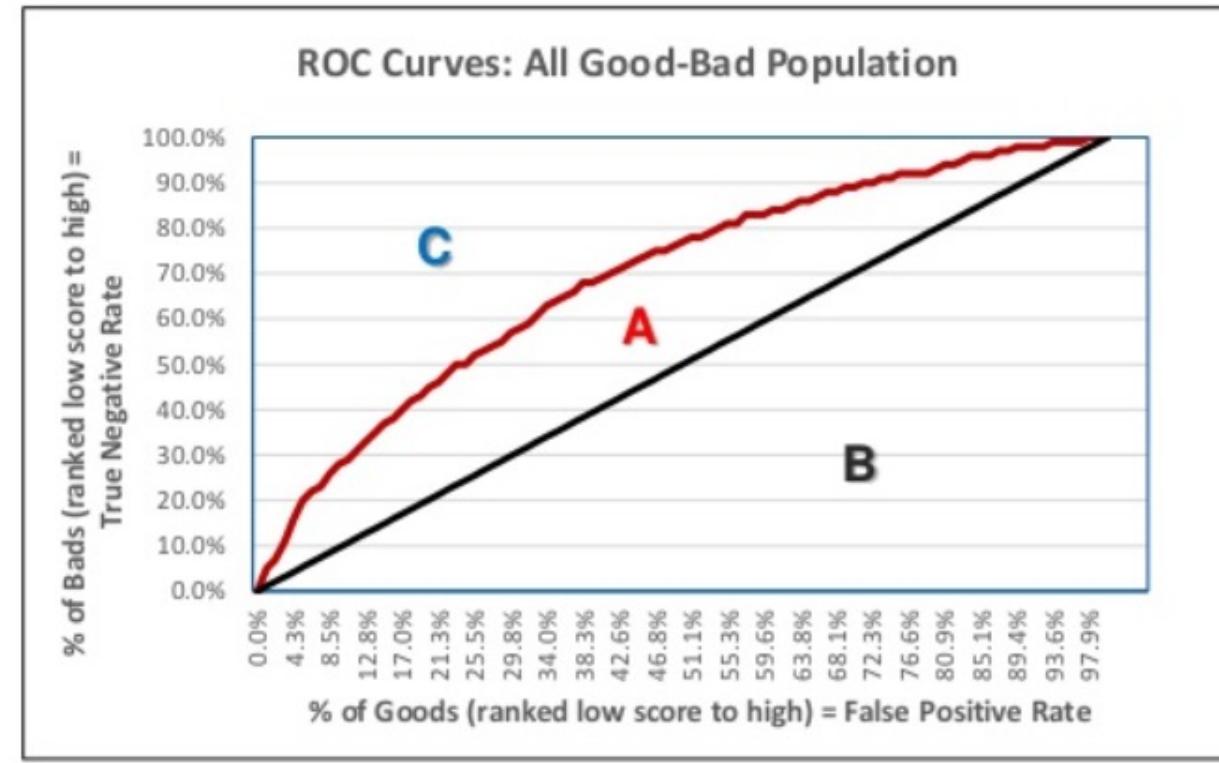
Accuracy	$= \frac{TN+TP}{TN+FP+FN+TP}$
Precision	$= \frac{TP}{FP+TP}$
Sensitivity	$= \frac{TP}{TP+FN}$
Specificity	$= \frac{TN}{TN+FP}$

Receiver operating characteristic (ROC)

ROC CURVE EXAMPLES



- The best classification has the largest area under the curve.
- Too sensitive to errors in the "gold standard" classification.



$$AUC = \text{Area A} + \text{Area B}$$

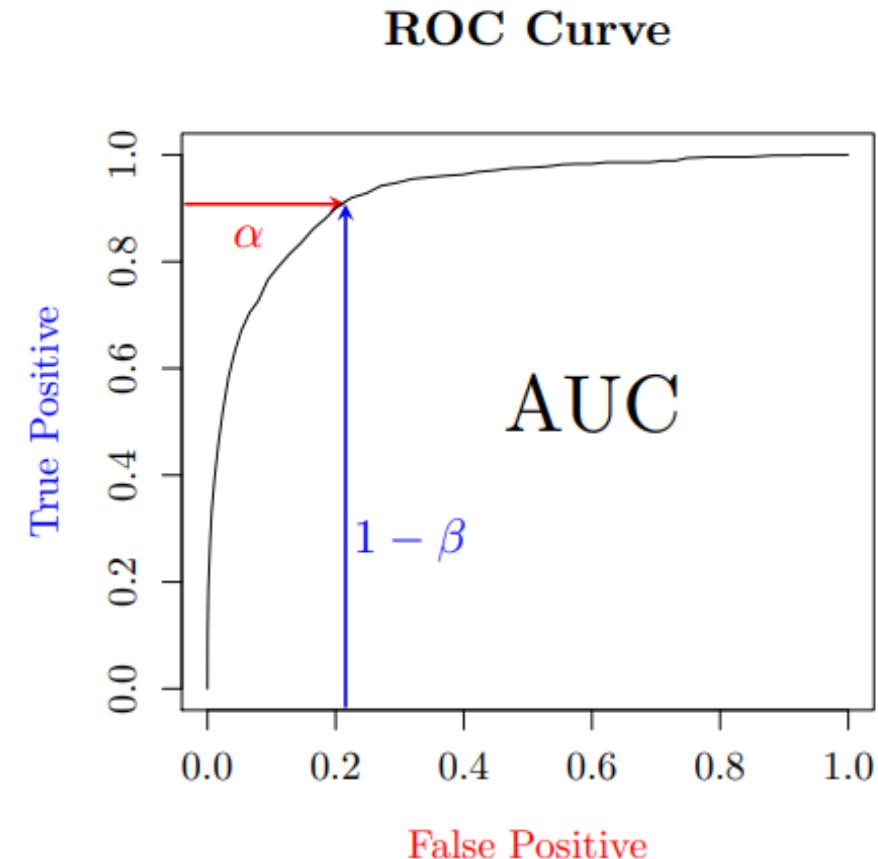
$$\text{Gini Coefficient} = 2 \times AUC - 1$$

Receiver operating characteristic (ROC)

	Sick	Healthy
Treating as sick	True Positive	False Positive
Treating as healthy	False Negative	True Negative

$$\text{Power} = \frac{TP}{TP + FN} = 1 - \beta$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} = \alpha$$



Contoh 2: Latihan dengan R

German Credit Data Set

The German Credit Data contains data on 20 variables and the classification whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants.

The data contains 1000 observations (700 good loans, 300 bad loans)

A predictive model developed on this data is expected to provide a bank manager guidance for making a decision whether to approve a loan to a prospective applicant based on his/her profiles.

```
#Membaca data
#Membaca data
install.packages("fairml")
library(fairml)

data(german.credit)
str(german.credit)

str(germancredit)
> str(german.credit)
'data.frame': 1000 obs. of 21 variables:
 $ Account_status      : Factor w/ 4 levels "< 0 DM",">= 200 DM",...: 1 3 4 ...
 $ Duration             : num  6 48 12 42 24 36 24 36 12 30 ...
 $ Credit_history       : Factor w/ 5 levels "all credits at this bank paid ...
 $ Purpose              : Factor w/ 10 levels "business","car (new)",...: 8 8...
 $ Credit_amount         : num  1169 5951 2096 7882 4870 ...
 $ Savings_bonds        : Factor w/ 5 levels "< 100 DM",">= 1000 DM",...: 5 1...
 $ Present_employment_since: Factor w/ 5 levels "< 1 year",">= 7 years",...: 2 3...
 $ Installment_rate     : num  4 2 2 2 3 2 3 2 2 4 ...
 $ Other_debtors_guarantors: Factor w/ 3 levels "co-applicant",...: 3 3 3 2 3 3 ...
 $ Resident_since        : num  4 2 3 4 4 4 4 2 4 2 ...
 $ Property              : Factor w/ 4 levels "building society savings agree...
 $ Age                   : num  67 22 49 45 53 35 53 35 61 28 ...
 $ Other_installment_plans: Factor w/ 3 levels "bank","none",...: 2 2 2 2 2 2 ...
 $ Housing               : Factor w/ 3 levels "rent","own","for free": 2 2 2 ...
 $ Existing_credits      : num  2 1 1 1 2 1 1 1 1 2 ...
 $ Job                   : Factor w/ 4 levels "management / self-employed / h...
 $ People_maintenance_for: num  1 1 2 2 2 2 1 1 1 1 ...
 $ Telephone              : Factor w/ 2 levels "none","yes": 2 1 1 1 1 2 1 2 1 ...
 $ Foreign_worker         : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 ...
 $ Credit_risk            : Factor w/ 2 levels "BAD","GOOD": 2 1 2 2 1 2 2 2 2 ...
 $ Gender                : Factor w/ 2 levels "Female","Male": 1 2 1 1 1 1 1 ...
```

Peubah:

1. Account_status: a factor with four levels representing the amount of money in the account or "no chcking account".
2. Duration: a continuous variable, the duration in months.
3. Credit_history: a factor with five levels representing possible credit history backgrounds.
4. Purpose: a factor with ten levels representing possible reasons for taking out a loan.
5. Credit_amount: a continuous variable.
6. Savings_bonds: a factor with five levels representing amount of money available in savings and bonds or "unknown / no savings account".
7. Present_employment_since: a factor with five levels representing the length of tenure in the current employment or "unemployed".
8. Installment_rate: a continuous variable, the installment rate in percentage of disposable income.
9. Other_debtors_guarantors: a factor with levels "none", "co-applicant" and "guarantor".
10. Resident_since: a continuous variable, number of years in the current residence.
11. Property: a factor with four levels describing the type of property to be bought or "unknown / no property".
12. Age: a continuous variable, the age in years.
13. Other_installment_plans: a factor with levels "bank", "none" and "stores".
14. Housing: a factor with levels "rent", "own" and "for free".
15. Existing_credits: a continuous variable, the number of existing credit lines at this bank.
16. Job: a factor with four levels for different job descriptions.
17. People_maintenance_for: a continuous variable, the number of people being liable to provide maintenance for.
18. Telephone: a factor with levels "none" and "yes".
19. Foreign_worker: a factor with levels "no" and "yes".
20. Credit_risk: a factor with levels "BAD" and "GOOD". → Response variable
21. Gender: a factor with levels "Male" and "Female".

```

##Konstruksi model dengan data training 80%, dan data testing 20%
library(caret)
set.seed(12420246)
in.train <- createDataPartition(as.factor(german.credit$Credit_risk), p=0.8, list=FALSE)
german.credit.train <- german.credit[in.train,]
german.credit.test <- german.credit[-in.train,]

credit.glm0 <- glm(Credit_risk ~ ., family = binomial, german.credit.train)
credit.glm.step <- step(credit.glm0)
credit.glm.step$anova

summary(credit.glm.step)

```

```

Step: AIC=776.26
Credit_risk ~ Account_status + Duration + Credit_history + Purpose +
  Credit_amount + Savings_bonds + Present_employment_since +
  Installment_rate + Other_debtors_guarantors + Other_installment_plans +
  Housing + Foreign_worker + Gender

          Df Deviance    AIC
<none>      704.26 776.26
- Gender       1   706.37 776.37
- Other_debtors_guarantors  2   709.13 777.13
- Housing       2   709.65 777.65
- Credit_amount  1   708.79 778.79
- Other_installment_plans  2   712.36 780.36
- Foreign_worker  1   711.15 781.15
- Present_employment_since 4   717.36 781.36
- Installment_rate  1   711.82 781.82
- Duration       1   714.48 784.48
- Credit_history  4   726.07 790.07
- Savings_bonds  4   726.68 790.68
- Purpose        9   738.98 792.98
- Account_status 3   758.74 824.74

```

```

> credit.glm.step$anova
           Step Df Deviance Resid. Df Resid. Dev      AIC
                           NA      NA     753 697.4557 791.4557
2                  - Property  3 1.2504839  756 698.7062 786.7062
3                  - Job      3 2.2388200  759 700.9450 782.9450
4                  - Telephone 1 0.4198912  760 701.3649 781.3649
5 - People_maintenance_for 1 0.5168207  761 701.8817 779.8817
6      - Resident_since    1 0.5915425  762 702.4732 778.4732
7                  - Age      1 0.8074193  763 703.2807 777.2807
8      - Existing_credits  1 0.9775937  764 704.2583 776.2583

```

```
## Final model  
credit.glm.final <- glm(Credit_risk ~ Account_status + Duration + Credit_history + Purpose +  
    Credit_amount + Savings_bonds + Present_employment_since +  
    Installment_rate + Other_debtors_guarantors + Other_installment_plans +  
    Housing + Foreign_worker + Gender, family = binomial, german.credit.train)  
  
summary(credit.glm.final)
```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 977.38 on 799 degrees of freedom
Residual deviance: 704.26 on 764 degrees of freedom
AIC: 776.26

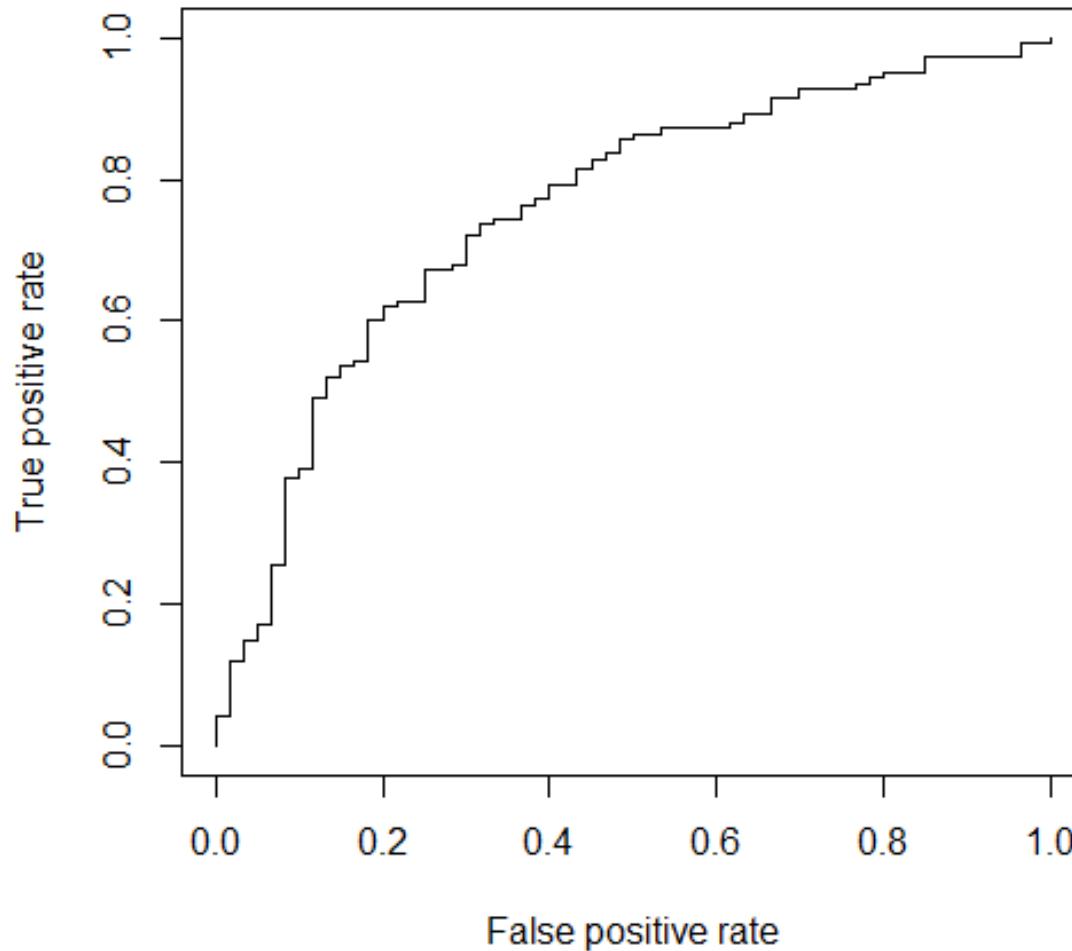
Number of Fisher Scoring iterations: 5

```
##Evaluation
#ConfusionMatrix
fit.final <- fitted.values(credit.glm.final)
pred.final <- ifelse(fit.final>=0.5, "GOOD", "BAD")

tab <- table(german.credit.train$Credit_risk,pred.final, dnn = c("Truth", "Predicted"))
tab
acc <- sum(diag(tab))/sum(tab)
acc
```

```
> tab
      Predicted
      BAD GOOD
Truth
  BAD 132 108
  GOOD 56 504
> acc
[1] 0.795
```

```
install.packages("ROCR")
library(ROCR)
pred<-
prediction(predict.glm(credit.glm.final,german.credit.test),german.credit.test$Credit_risk)
perf <- performance(pred,"tpr","fpr")
plot(perf)
```



```
AUC.final<-performance(pred, measure = "auc")@y.values[[1]]  
AUC.final  
[1] 0.7552381
```

Tugas Kelompok – Sesi UTs

- Buatlah proposal penelitian mengenai Projek Kelompok-nya, yang di dalamnya berisi
 1. Judul berdasarkan topik projek yang ditentukan
 2. Latar belakang dan tujuan
 3. Data dan peubah-peubah yang digunakan
 4. Metodologi (rencana tahapan analisis data yang dilakukan)
- Selain proposal penelitian dalam format makalah, dikumpulkan juga file presentasi dalam powerpoint.
- File proposal penelitian dan file presentasi diupload pada form
<https://ipb.link/project-uts-tpm>
- Batas waktu pengiriman adalah **Hari Jumat, tanggal 3 Maret 2023 jam 23:59 WIB**

- Komponen Penilaian:
 - Kecepatan pengiriman
 - Kesesuaian isi proposal
 - Orisinalitas
- Selanjutnya, pada pertemuan 7, akan diadakan **Sesi Presentasi** sesuai dengan file presentasi yang dikirimkan, dengan aturan:
 - Kelompok dipilih secara acak
 - Penilaian sesi presentasi ini berdasarkan keaktifan dan kesesuaian pertanyaan/jawaban setiap mahasiswa pada forum diskusi yang ada

Topik Projek Kelompok

1. Topik 1: Supervised learning dengan peubah respon numerik
2. Topik 2: Supervised learning dengan peubah respon kategorik
3. Topik 3: Unsupervised learning dengan kasus penggerombolan
4. Topik 4: Unsupervised learning dengan kasus reduksi dimensi

Terima kasih 😊



Classification Tree CART

Kuliah 4 - STA1382 Teknik
Pembelajaran Mesin

Septian Rahardiantoro



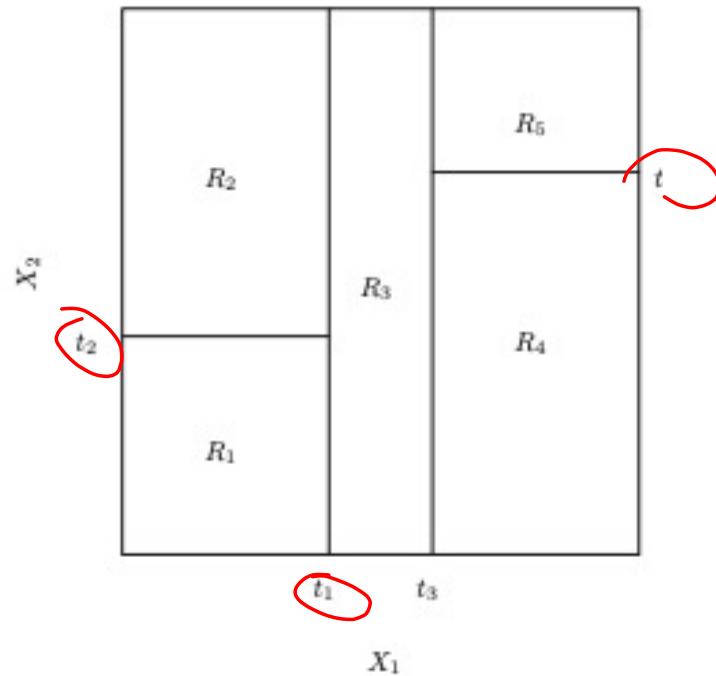
Outline

- Overview Decision Tree
- Classification Tree CART
 - 1. Pengenalan Konsep Entropy dan Information Gain
 - 2. Pengenalan Algoritma Dasar Pohon Klasifikasi
 - 3. Menilai Kemampuan Prediksi Pohon Klasifikasi

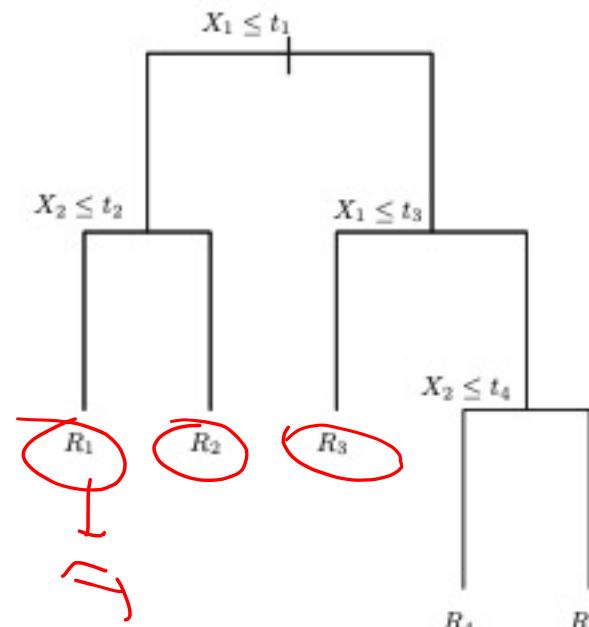
Overview Decision Tree

- Untuk membuat prediksi untuk pengamatan tertentu, biasanya digunakan rata-rata pengamatan data training di wilayah yang menjadi miliknya.
- Karena seperangkat aturan pemisahan yang digunakan untuk mensegmentasi ruang prediktor dapat dirangkum dalam sebuah pohon, jenis pendekatan ini dikenal sebagai metode pohon keputusan.
- Keunggulan metode berbasis pohon adalah bersifat sederhana dan berguna untuk interpretasi. ✓

- Pohon keputusan (decision tree) dapat diterapkan pada masalah regresi dan klasifikasi
- Regression Tree → tree untuk peubah Y numerik



$$Y \sim X_1 + X_2$$



- Setelah daerah R_1, R_2, \dots, R_J telah dibuat, peubah respon diprediksi dengan menggunakan rata-rata pengamatan pada wilayah tempat pengamatan uji itu berada.

Classification Tree CART

- Merupakan pohon keputusan yang diaplikasikan pada kasus regresi dengan peubah Y berskala kategorik
- Seringkali disebut dengan istilah:
 - Classification Tree
 - Decision Tree
 - Recursive Partition
 - Iterative Dichotomiser

Kegunaan

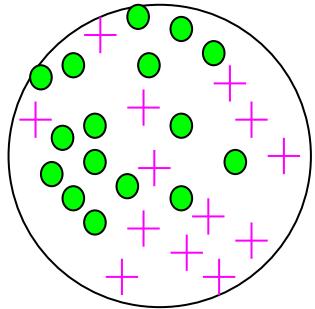
- Mengidentifikasi variabel apa yang dapat dijadikan sebagai pembeda antar kelompok
- Memprediksi keanggotaan kelompok suatu individu berdasarkan karakteristiknya
- Terapannya antara lain:
 - Marketing: Mengidentifikasi prospective customer (cross-sell, up-sell, new acquisition)
 - Risk: Credit scoring, 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 yang setara kegunaannya: Regresi Logistik, k-Nearest Neighbor, Discriminant Analysis, Support Vector Machine, Bayesian Classifier, dll

1. Pengenalan Entropy dan Information Gain

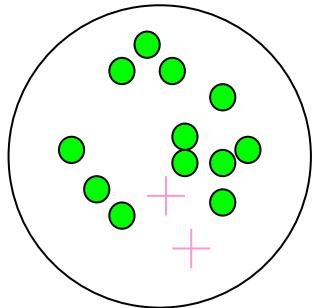
Entropy

- Andaikan sebuah gugus data D berisi individu-individu dengan dua kelas yaitu kelas YES dan NO, dengan proporsi yang YES sebesar p , dan tentu saja $(1 - p)$ 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 keheterogenan data (impurity)
- $E(D) = -p \log_2(p) - (1-p) \log_2(1-p)$
= 0

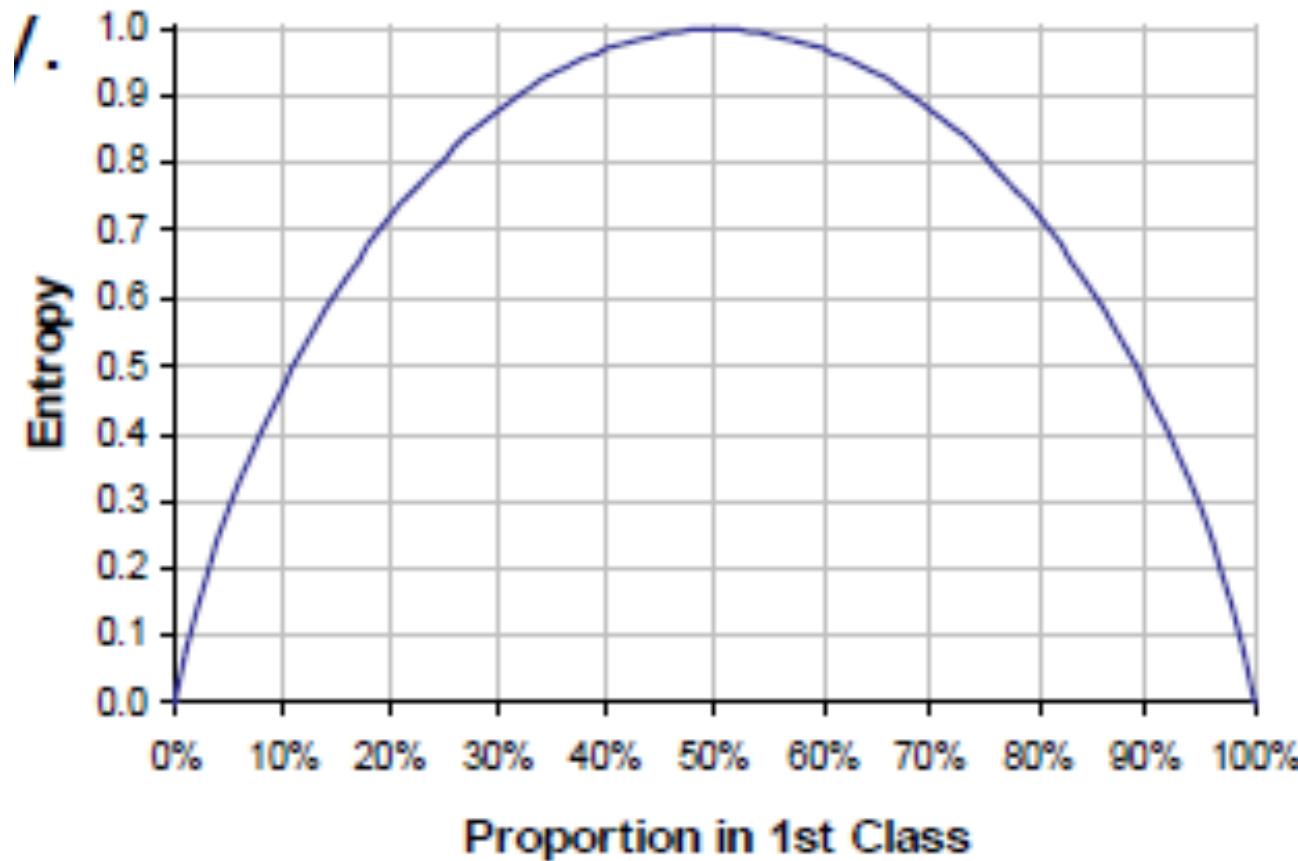
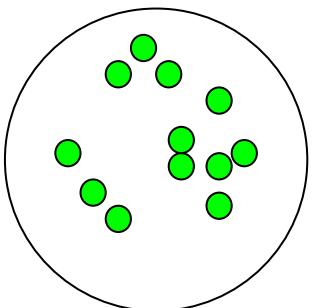
Very impure group



Less impure



Minimum impurity



Entropi

$$E(D) = -p \log_2(p) - (1-p) \log_2(1-p)$$

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

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

- Pemisahan yang menghasilkan kelompok-kelompok yang homogen → memiliki information gain yang semakin besar

Entropy & Information Gain

Frequency
Percent
Row Pct
Col Pct

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

E(D)

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

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

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

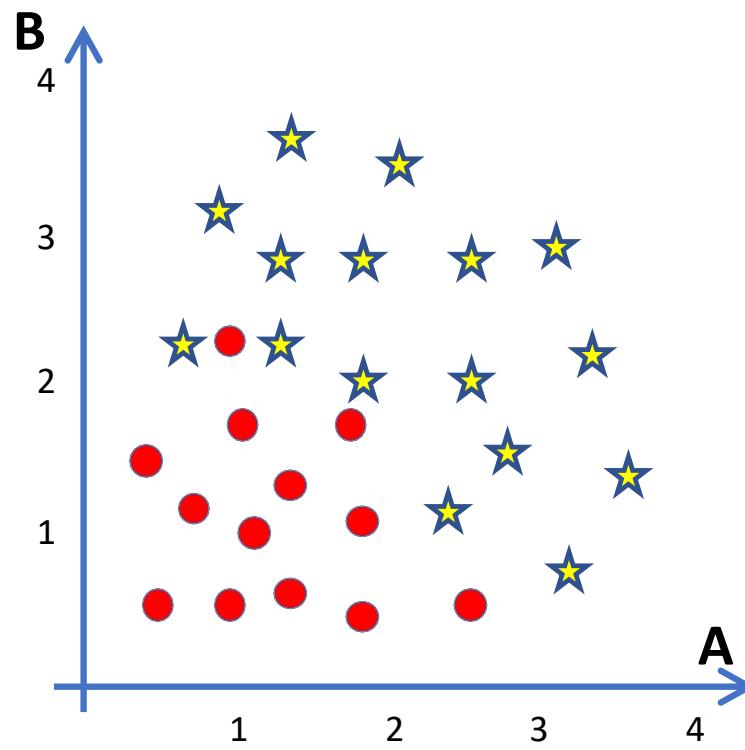
$$IG(D,V) = E(D) - \sum_{i=1}^k \frac{|D_i|}{|D|} E(D_i)$$

Information Gain dari pemisahan berdasarkan Jenis Kelamin

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

2. Pengenalan Algoritma Dasar Pohon Klasifikasi

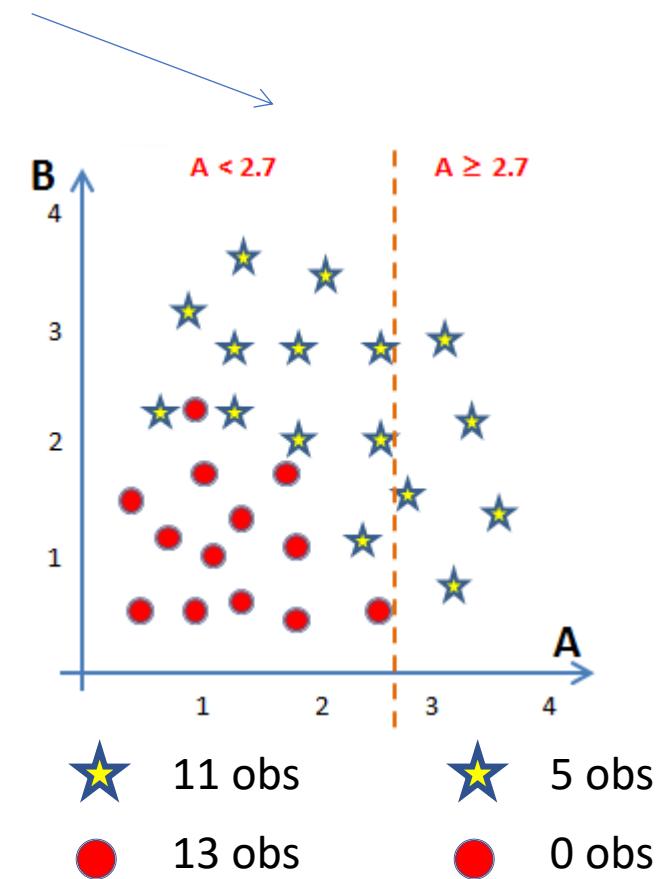
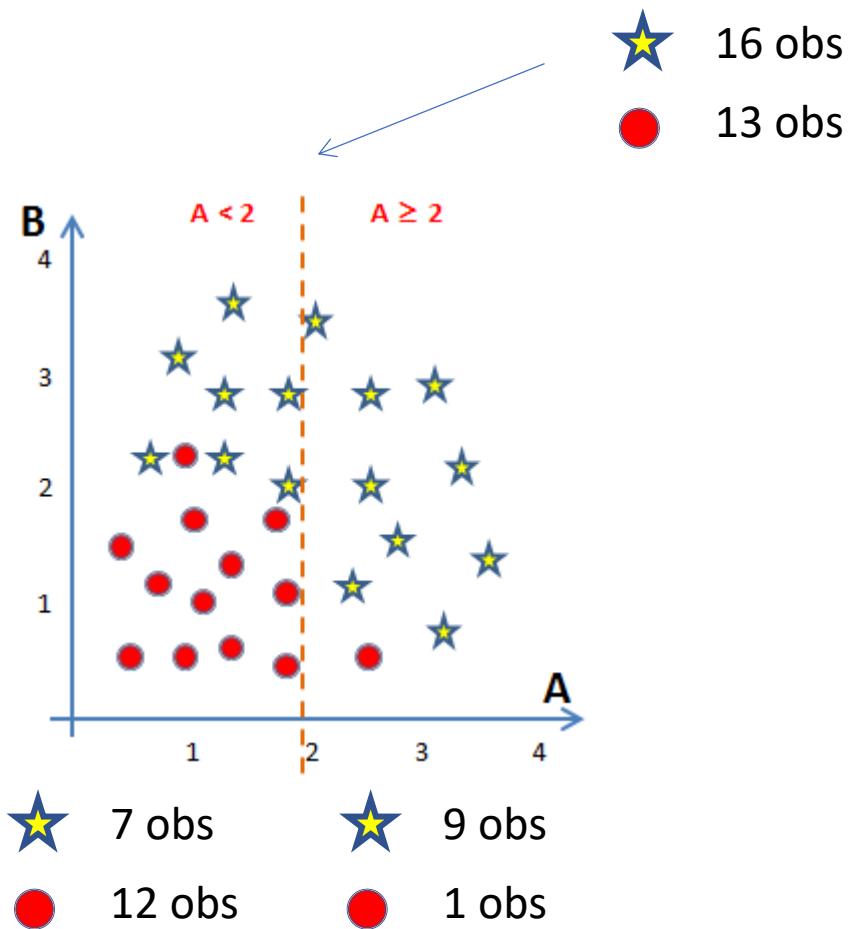
Ide Dasar



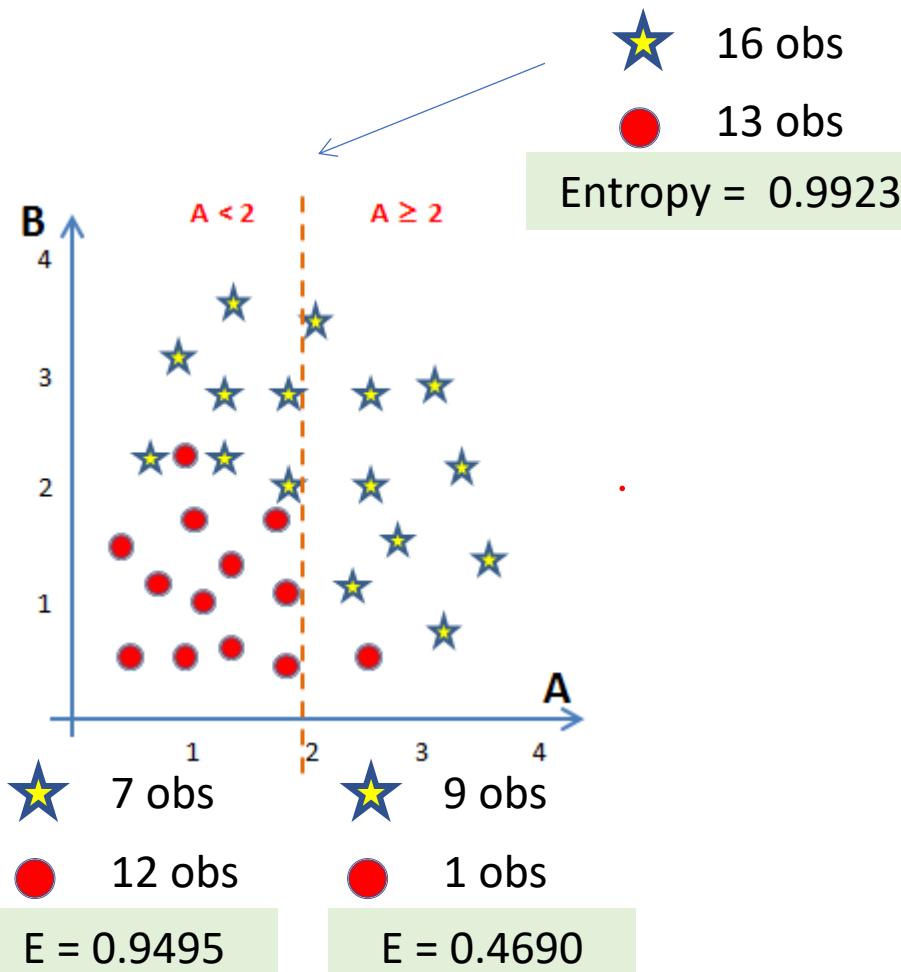
Mencari pemisah terbaik antara individu dengan individu

Pemisahan dilakukan untuk masing-masing variabel, bukan kombinasinya.

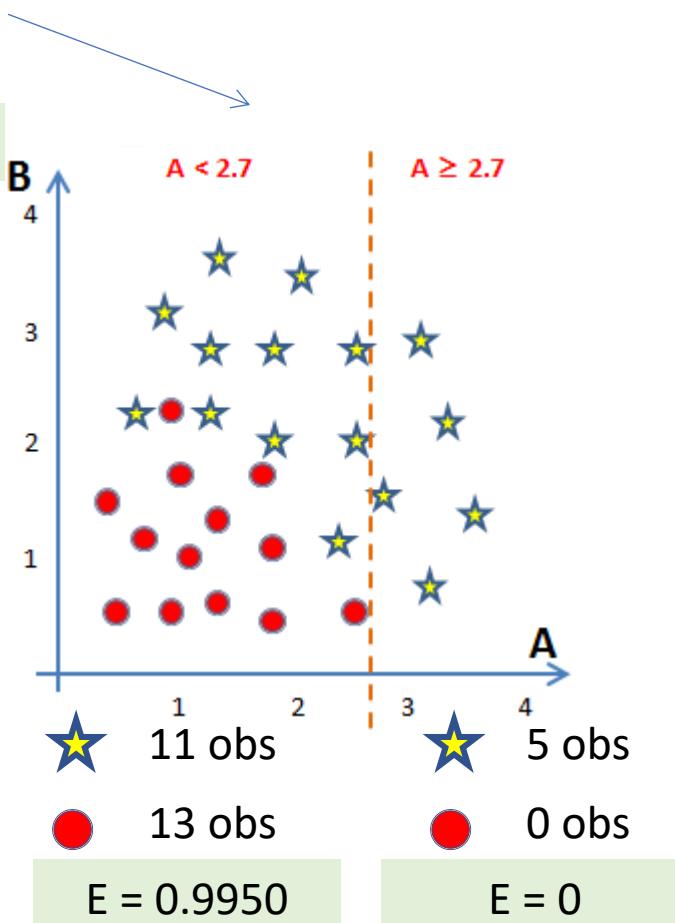
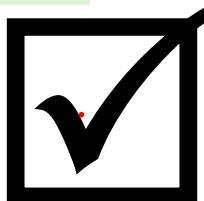
Pemisah yang dicari adalah yang menyebabkan data hasil pemisahannya bersifat homogen kelasnya.



Mana yang lebih baik?

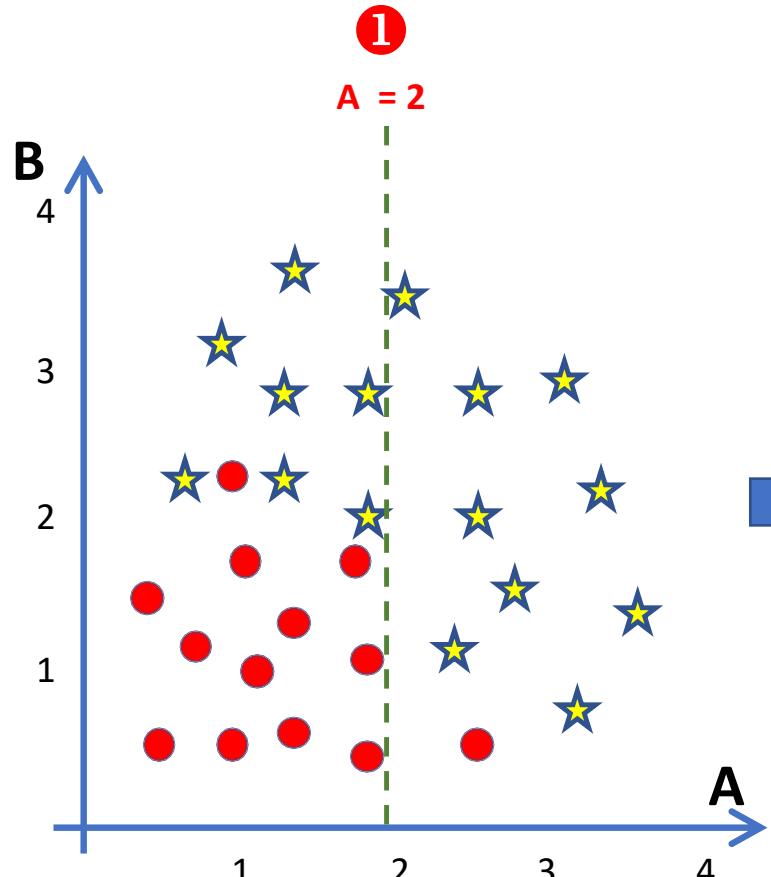


Information Gain =
0.2085

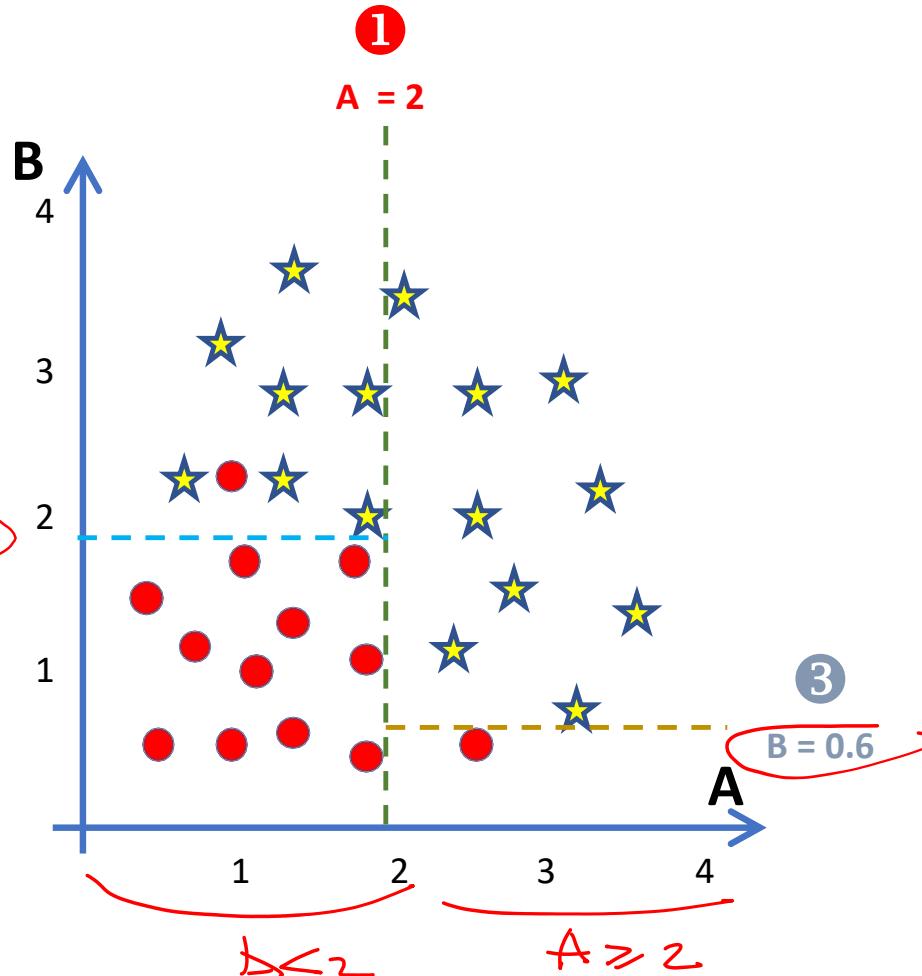


Information Gain =
0.1688

Ide Dasar: Tahapan



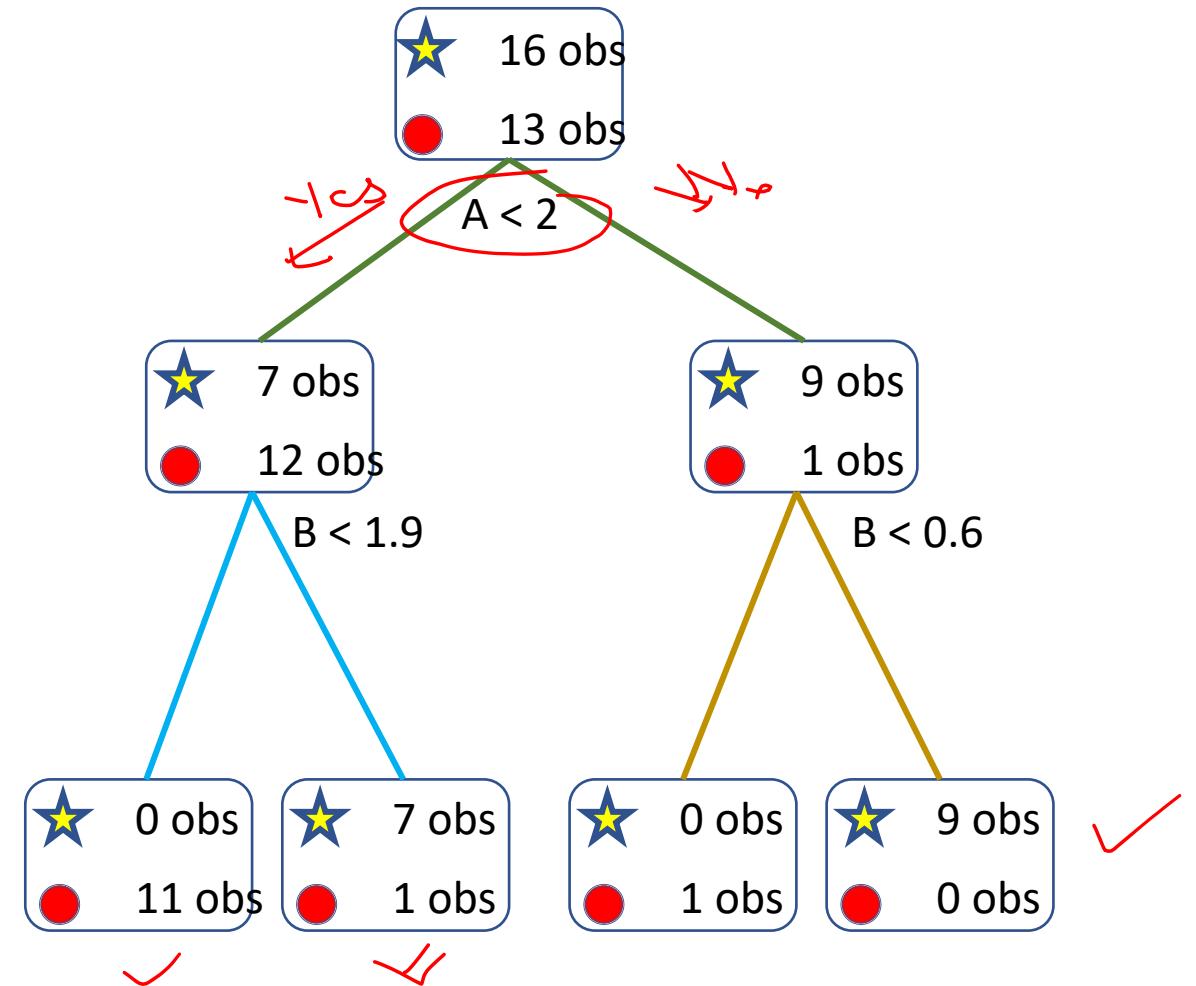
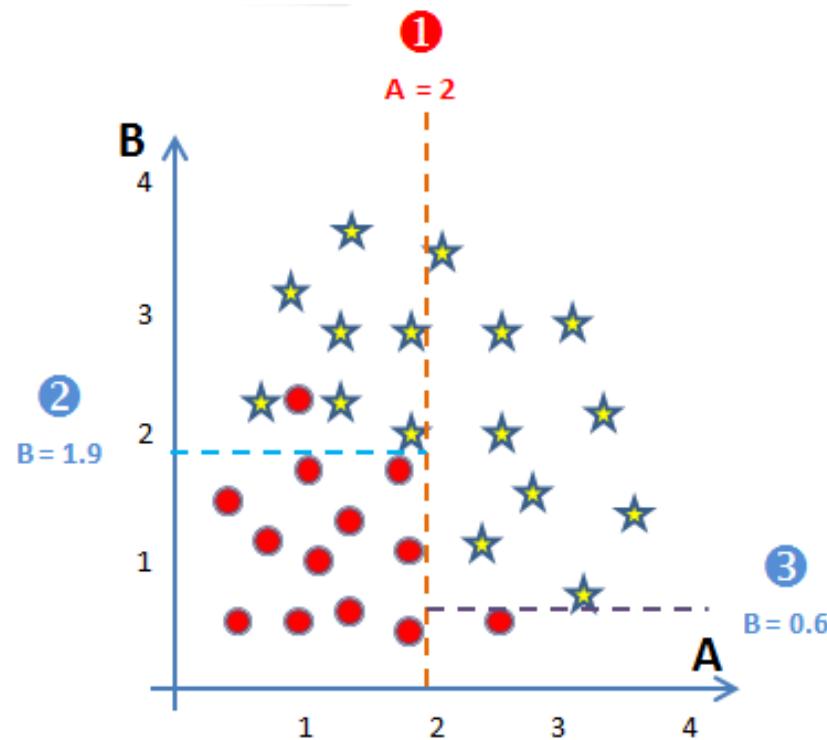
2
 $B = 1.9$



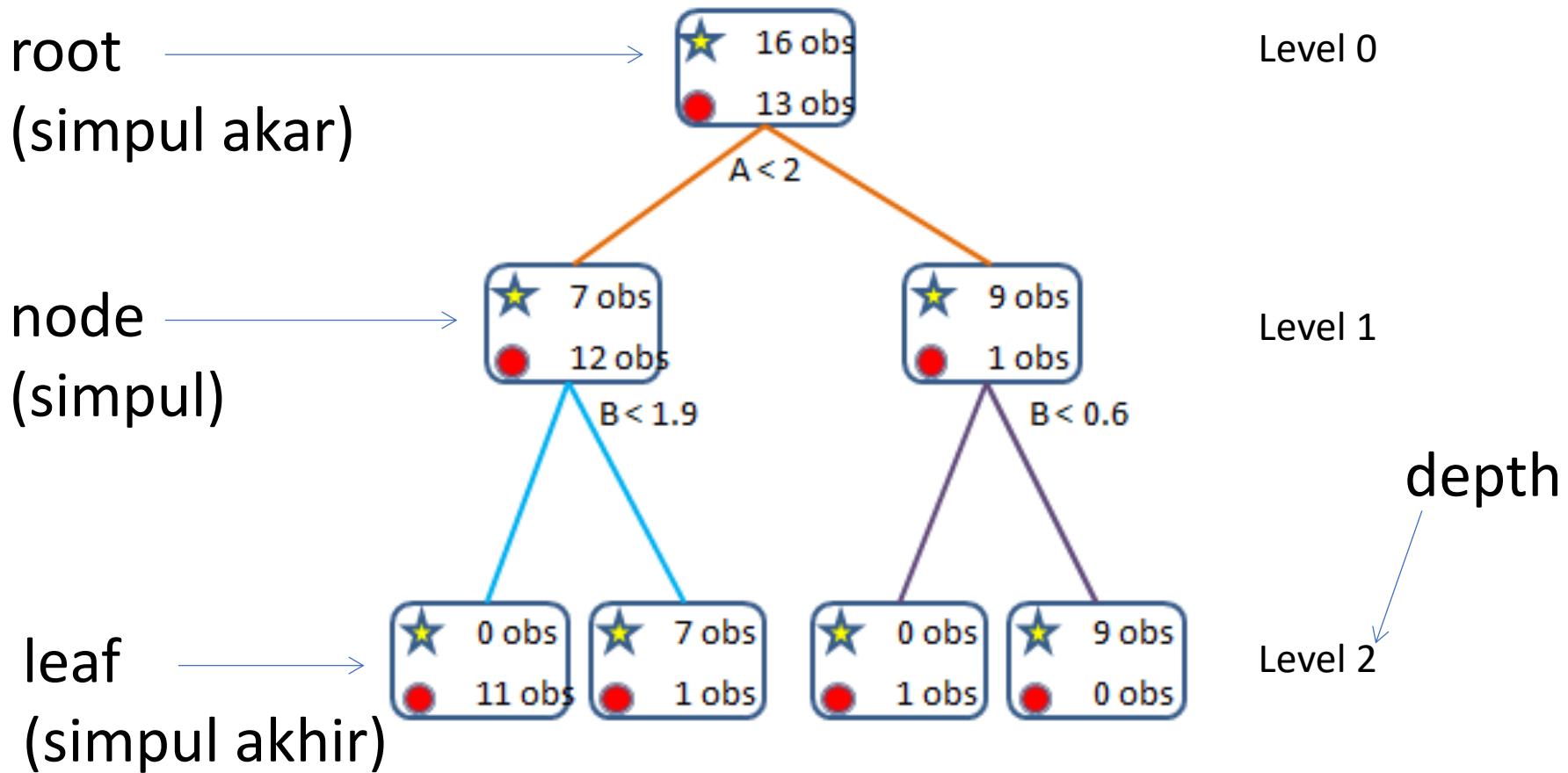
3
 $B = 0.6$

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

Representasi Hasil Pemisahan

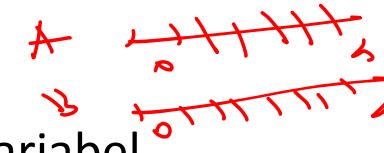


Beberapa istilah



Algoritma Dasar 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

- Proses pemisahan akan berhenti dengan kriteria:
 1. Simpul berisi amatan yang berasal dari satu kelas variabel respon
 2. Simpul berisi amatan yang seluruh variabel prediktornya identik
 3. Simpul berisi amatan yang kurang dari ukuran simpul minimal yang ditentukan di awal
 4. Kedalaman pohon sudah mencapai kedalaman maksimal

Ilustrasi Sederhana

- Gunakan “datatree01.csv”
- Variabel respon:
 - Tertarik_Beli
- Variabel Prediktor:
 - Jenis_Kelamin
 - Single
 - Perokok
 - Tinggal
 - Usia
 - Budget

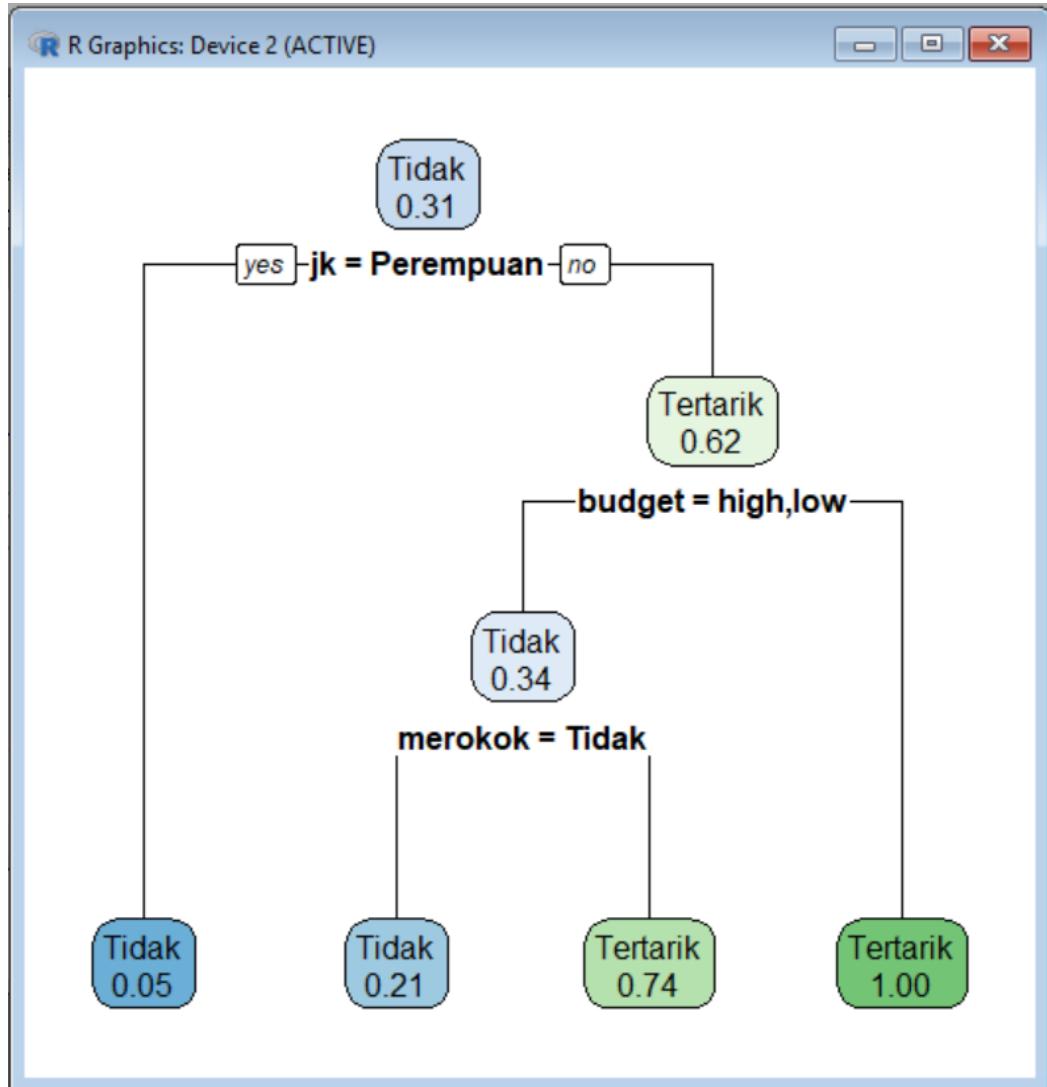
```
##Classification Tree

propensity <- read.csv("D:/datatree01.csv", sep=";", header=TRUE)
tertarik <- factor(propensity$Tertarik.Beli., levels = 0:1, labels = c("Tidak", "Tertarik"))
jk <- factor(propensity$Jenis.Kelamin,   levels = 0:1, labels = c("Perempuan", "Laki-Laki"))
kota <- factor(propensity$Tinggal.di.Kota, levels = 0:1, labels = c("Tidak", "Ya"))
single <- factor(propensity$Single, levels = 0:1, labels = c("Menikah", "Single"))
merokok <- factor(propensity$Perokok, levels = 0:1, labels = c("Tidak", "Ya"))
budget <- propensity$Budget
usia <- propensity$usia

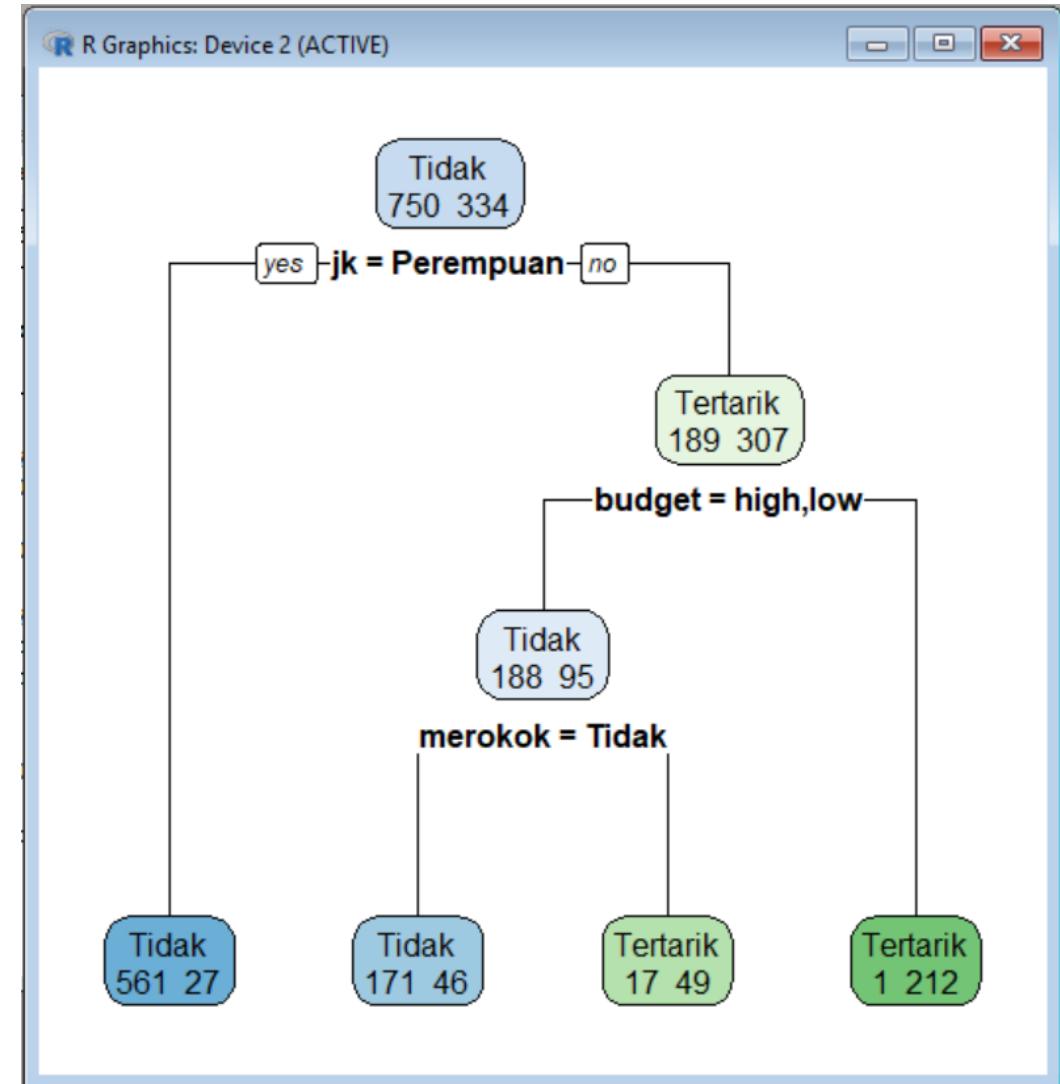
library(rpart)

model.01 <- rpart(tertarik ~  jk + kota + single + usia + merokok + budget,
                   method="class", control = rpart.control(minsplit = 100))
model.01
```

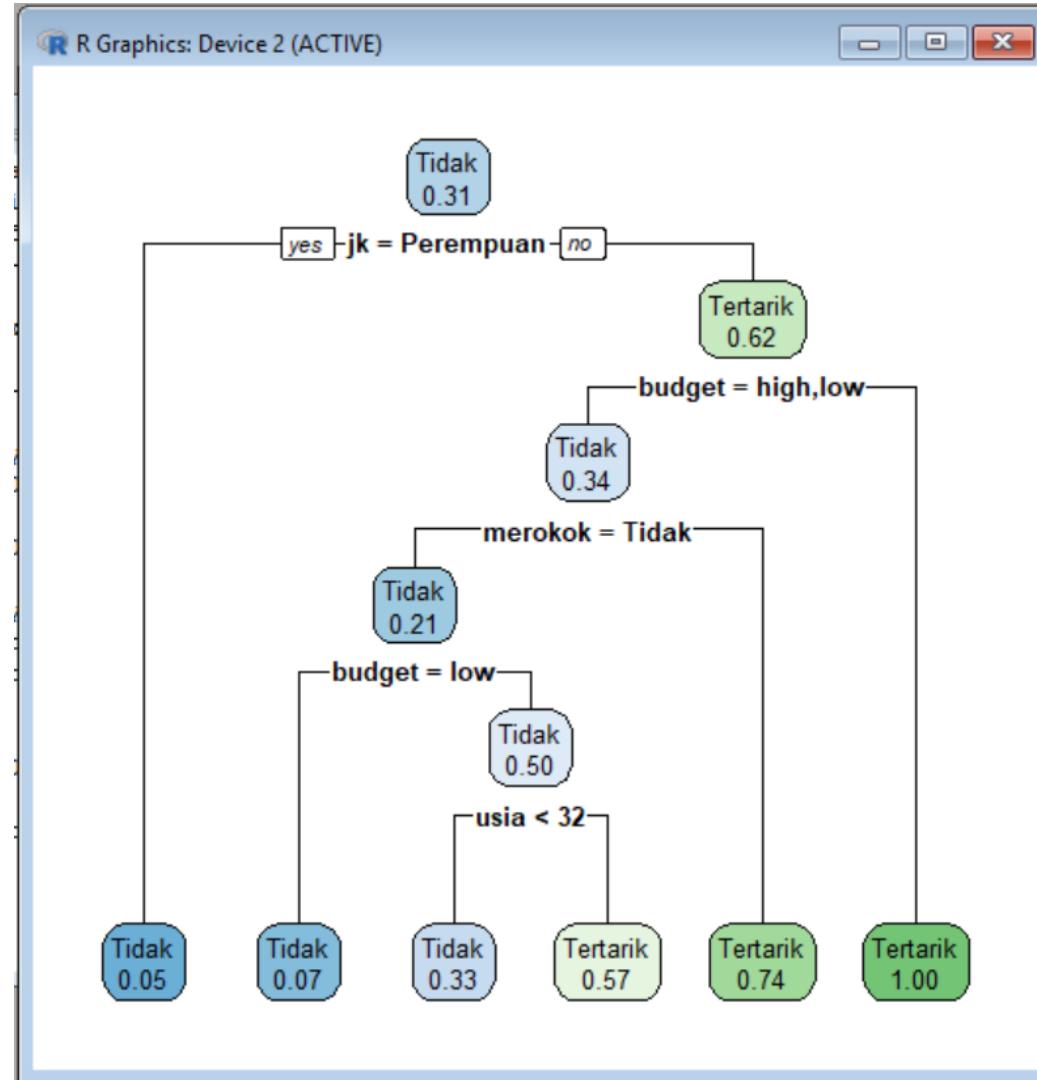
```
library(rpart.plot)
rpart.plot(model.01, extra=6)
```



```
rpart.plot(model.01, extra=1)
```



```
model.02 <- rpart(tertarik ~ jk + kota + single + usia + merokok + budget,  
                  method="class", control = rpart.control(minsplit = 50))  
rpart.plot(model.02, extra=6)
```



3. Menilai Kemampuan Prediksi Pohon Klasifikasi

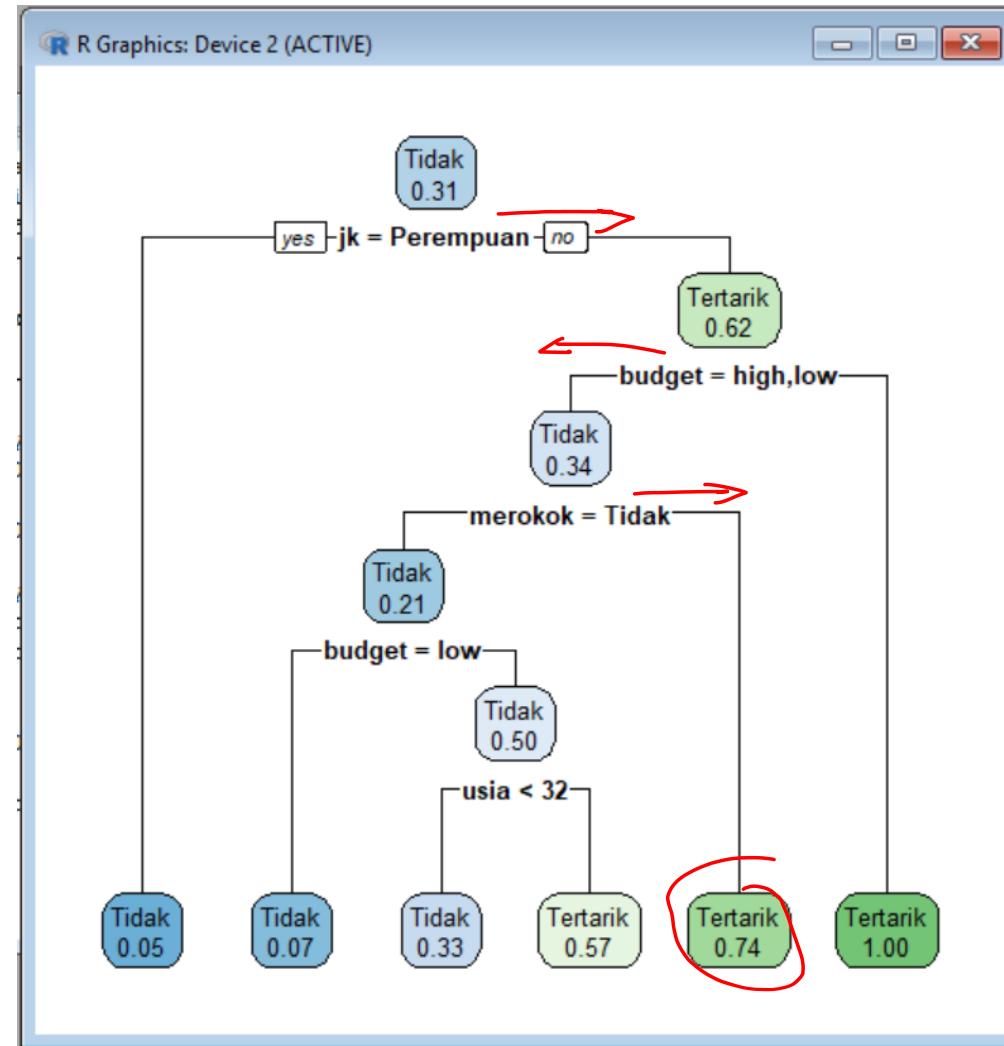
Prediksi Peubah Respon

- 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 Peubah Respon

Misal

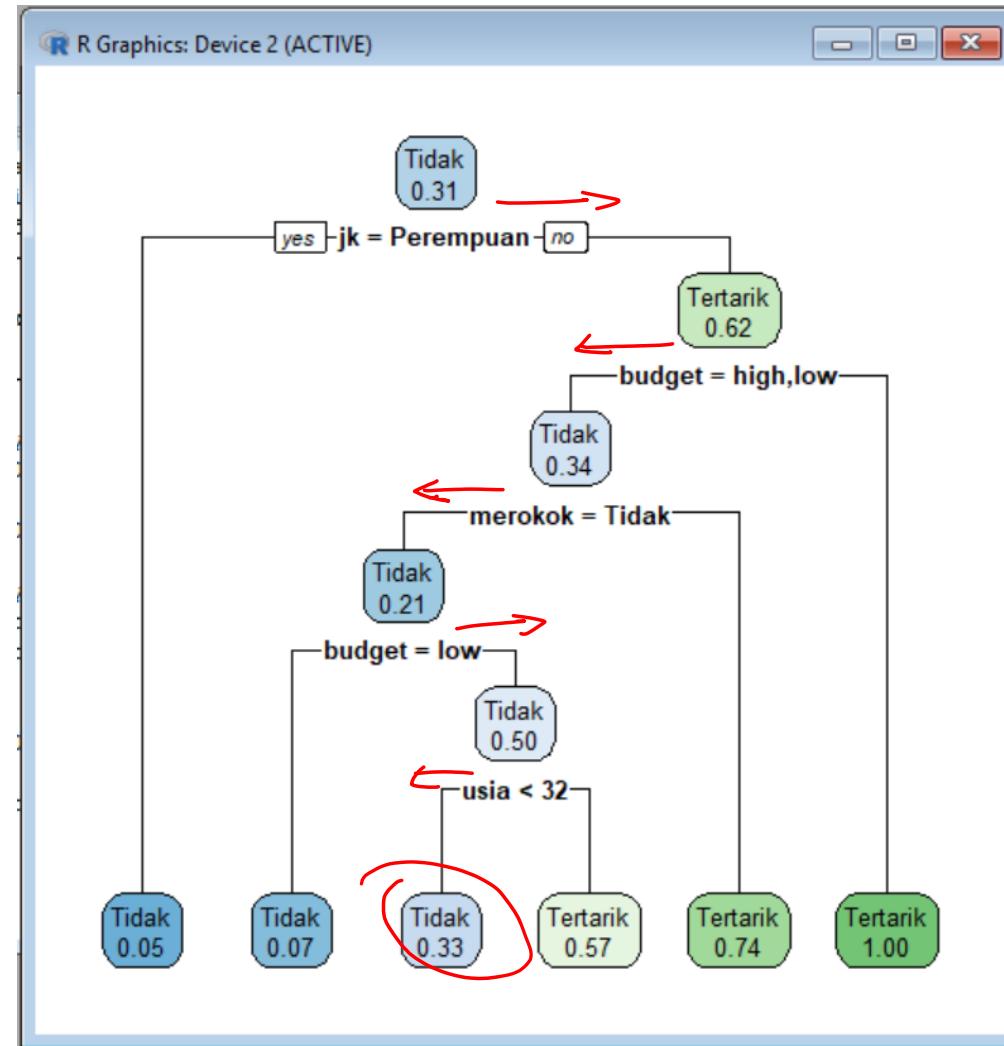
- Jenis Kelamin = Laki-Laki
 - Budget = Low ✓
 - Merokok = Ya
 - Usia 25 tahun
 - Tinggal di Kota
 - Single
- .
- Probability TERTARIK = 0.74



Prediksi Peubah Respon

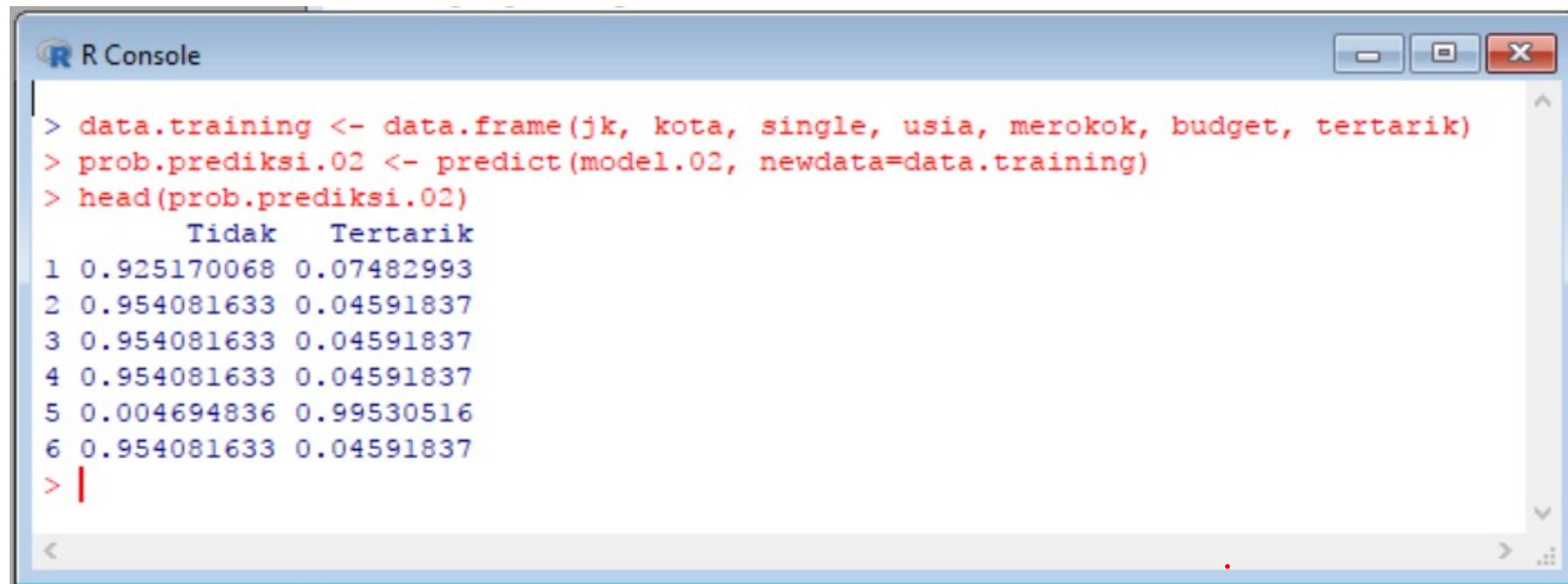
Misal

- Jenis Kelamin = Laki-Laki
 - Budget high ✓
 - Tidak Merokok
 - Usia 25 tahun ✓
 - Tinggal di Kota
 - Single
-
- Probability TERTARIK = 0.33



Prediksi Peubah Respon

```
data.training <- data.frame(jk, kota, single, usia, merokok, budget, tertarik)
prob.prediksi.02 <- predict(model.02, newdata=data.training)
head(prob.prediksi.02)
```



The screenshot shows an R console window titled "R Console". The window contains the following text:

```
> data.training <- data.frame(jk, kota, single, usia, merokok, budget, tertarik)
> prob.prediksi.02 <- predict(model.02, newdata=data.training)
> head(prob.prediksi.02)
  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
> |
```

Prediksi Peubah Respon

Andaikan digunakan batasan 0.5 untuk mengelompokkan ketertarikan, sehingga kalau

Prob(Tertarik) > 0.5 → Tertarik ✓

Prob(Tertarik) ≤ 0.5 → 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 ✓

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

Classification Table

		Predicted Class	
		0	1
Actual Class	0	True Negative	False Positive
	1	False Negative	True Positive
Predicted Negative		Predicted Positive	

Actual Negative

Actual Positive

Proporsinya harus tinggi

R Console

```
> confusionMatrix(prediksi.02, 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 : 0.9480

Specificity : 0.8653

Pos Pred Value : 0.9405

Neg Pred Value : 0.8811

Prevalence : 0.6919

Detection Rate : 0.6559

Detection Prevalence : 0.6974

Balanced Accuracy : 0.9066

'Positive' Class : Tidak

```
prediksi.02 <- factor(ifelse(prob.prediksi.02[,2] > 0.5, 1, 0),  
levels = 0:1, labels = c("Tidak", "Tertarik"))
```

```
library(caret)
```

```
confusionMatrix(prediksi.02, tertarik)
```

Goodness of Classification Tree

Andaikan batas peluangnya diganti dari **0.5** menjadi **0.6**

```
R Console

Confusion Matrix and Statistics

      Reference
Prediction Tidak Tertarik
 Tidak      732      73
 Tertarik    18     261

      Accuracy : 0.9161
      95% CI   : (0.8979, 0.9319)
 No Information Rate : 0.6919
 P-Value [Acc > NIR] : < 2.2e-16

      Kappa   : 0.7937
 Mcnemar's Test P-Value : 1.507e-08

      Sensitivity : 0.9760
      Specificity  : 0.7814
 Pos Pred Value  : 0.9093
 Neg Pred Value  : 0.9355
 Prevalence       : 0.6919
 Detection Rate  : 0.6753
 Detection Prevalence : 0.7426
 Balanced Accuracy : 0.8787

'Positive' Class : Tidak
```

```
prediksi.02 <- factor(ifelse(prob.prediksi.02[,2] > 0.6, 1, 0),
                      levels = 0:1, labels = c("Tidak", "Tertarik"))
```

```
library(caret)
confusionMatrix(prediksi.02, tertarik)
```

Goodness of Classification Tree

Andaikan batas peluangnya diganti dari **0.5** menjadi **0.3**

```
R Console
Confusion Matrix and Statistics

      Reference
Prediction Tidak Tertarik
 Tidak      697      38
Tertarik     53     296

      Accuracy : 0.9161
      95% CI  : (0.8979, 0.9319)
No Information Rate : 0.6919
P-Value [Acc > NIR] : <2e-16

      Kappa : 0.8055
McNemar's Test P-Value : 0.1422

      Sensitivity : 0.9293
      Specificity : 0.8862
      Pos Pred Value : 0.9483
      Neg Pred Value : 0.8481
      Prevalence : 0.6919
      Detection Rate : 0.6430
      Detection Prevalence : 0.6780
      Balanced Accuracy : 0.9078

'Positive' Class : Tidak
```

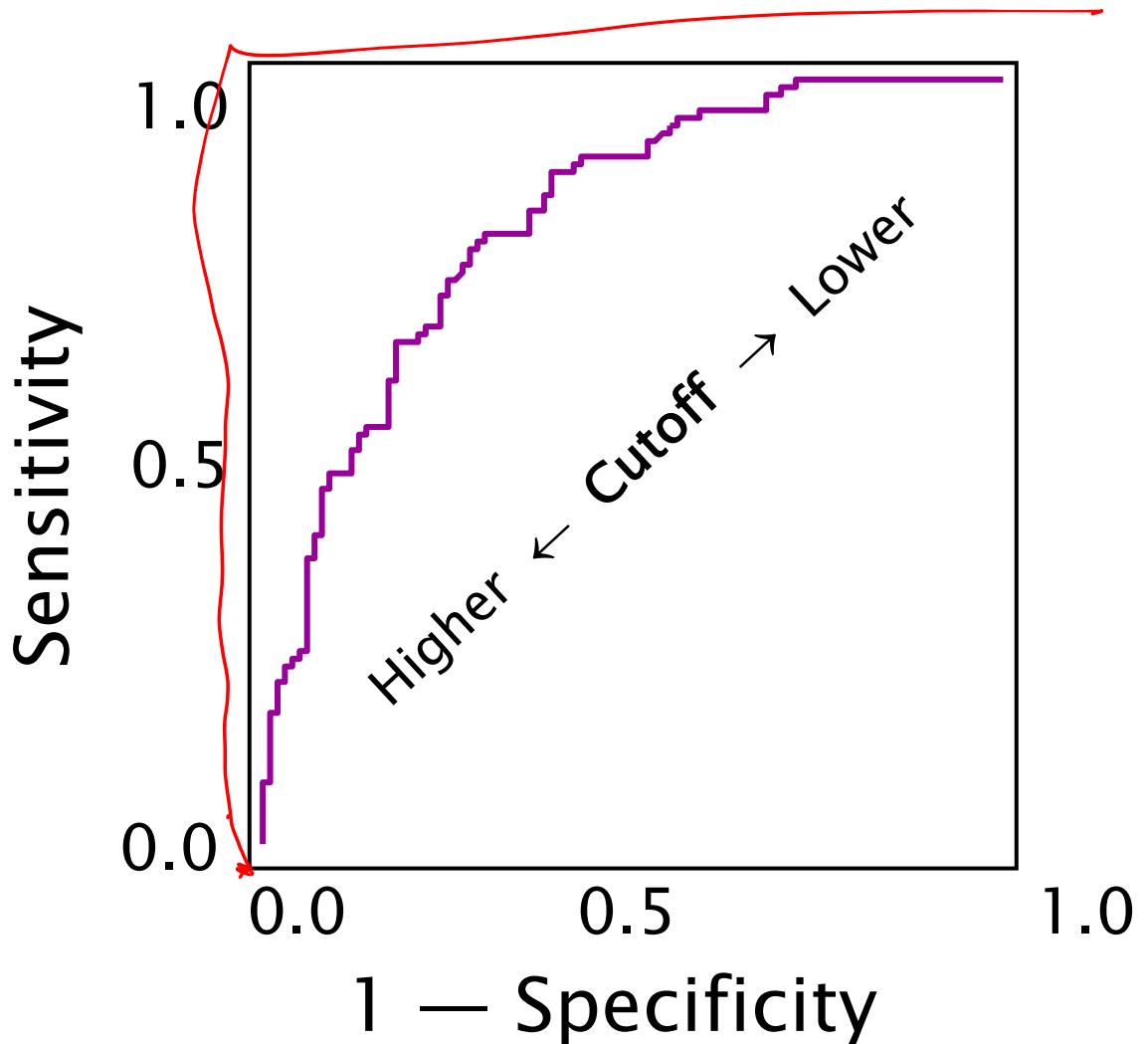
```
prediksi.02 <- factor(ifelse(prob.prediksi.02[,2] > 0.3, 1, 0),
levels = 0:1, labels = c("Tidak", "Tertarik"))

library(caret)
confusionMatrix(prediksi.02, tertarik)
```

Goodness of Classification Tree

Cut-Off	Accuracy	Sensitivity	Specificity
0.3	91.61%	92.93%	88.62%
0.5	92.25%	94.80%	86.53%
0.6	91.61%	97.60%	78.14%

ROC Curve



- Kurva ROC yang ideal adalah garis vertikal dari asal ke sensitivitas 1, dan kemudian garis horizontal sepanjang sensitivitas 1 untuk semua tingkat ($1 - \text{spesifisitas}$)
- Kurva ROC biasanya digunakan untuk membandingkan model pesaing
- Ukuran numerik seberapa dekat kurva ROC cocok dengan kurva ideal dihitung dengan membandingkan area di bawah kurva dengan 1. Ternyata luasnya sama dengan statistik c.

ROC Curve

```
library(ROCR)

pred <-
prediction(prob.prediksi.02[,1],
tertarik)

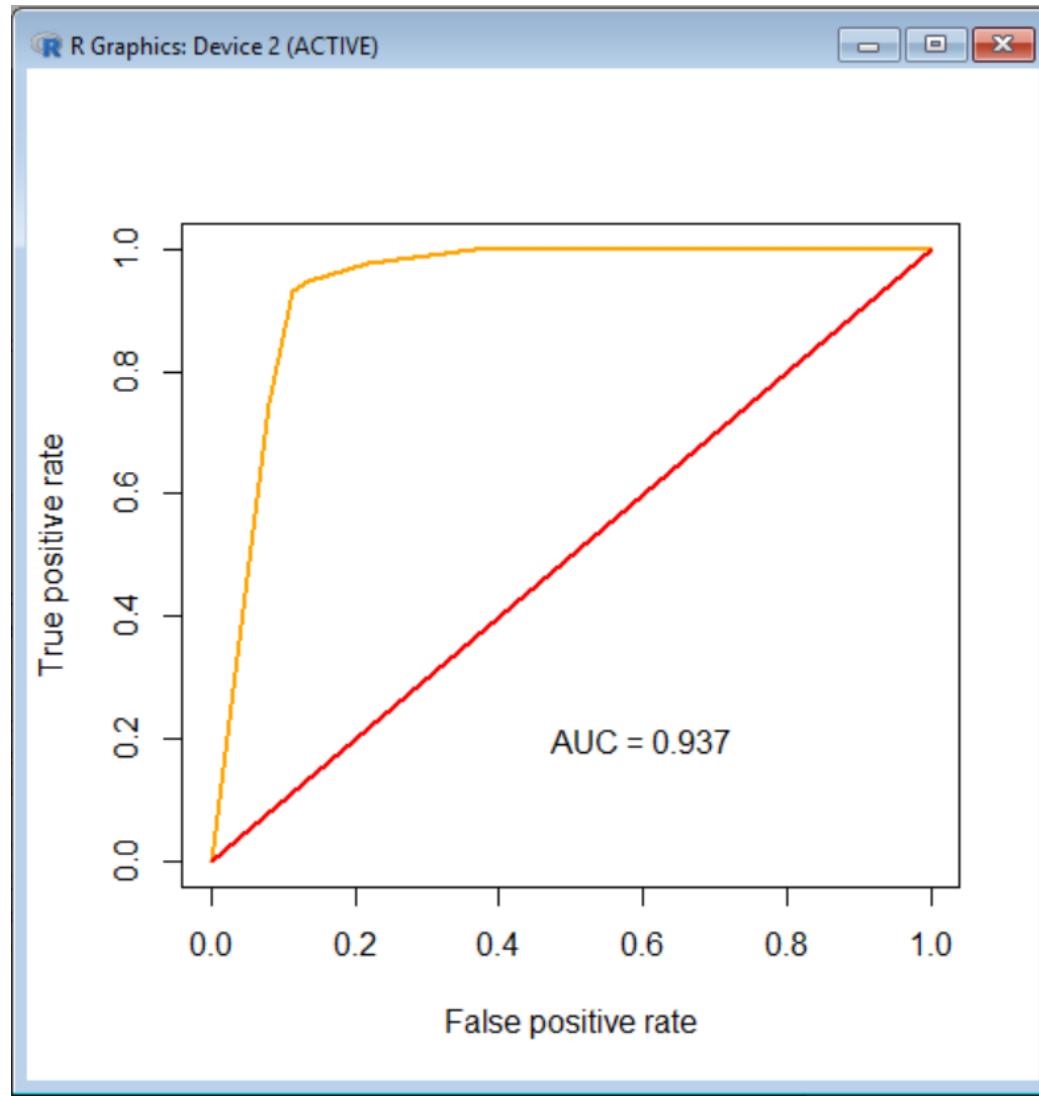
roc <- performance(pred,
measure="tpr", x.measure="fpr")

auc <- performance(pred, 'auc')

AUC <- auc@y.values[[1]]


plot(roc, col="orange", lwd=2)
lines(x=c(0, 1), y=c(0, 1),
col="red", lwd=2)

text(0.6, 0.2, paste0("AUC = ",
round(AUC,3)) )
```



Terima kasih 😊



Artificial Neural Network

Kuliah 5 - STA1382 Teknik
Pembelajaran Mesin

Septian Rahardiantoro

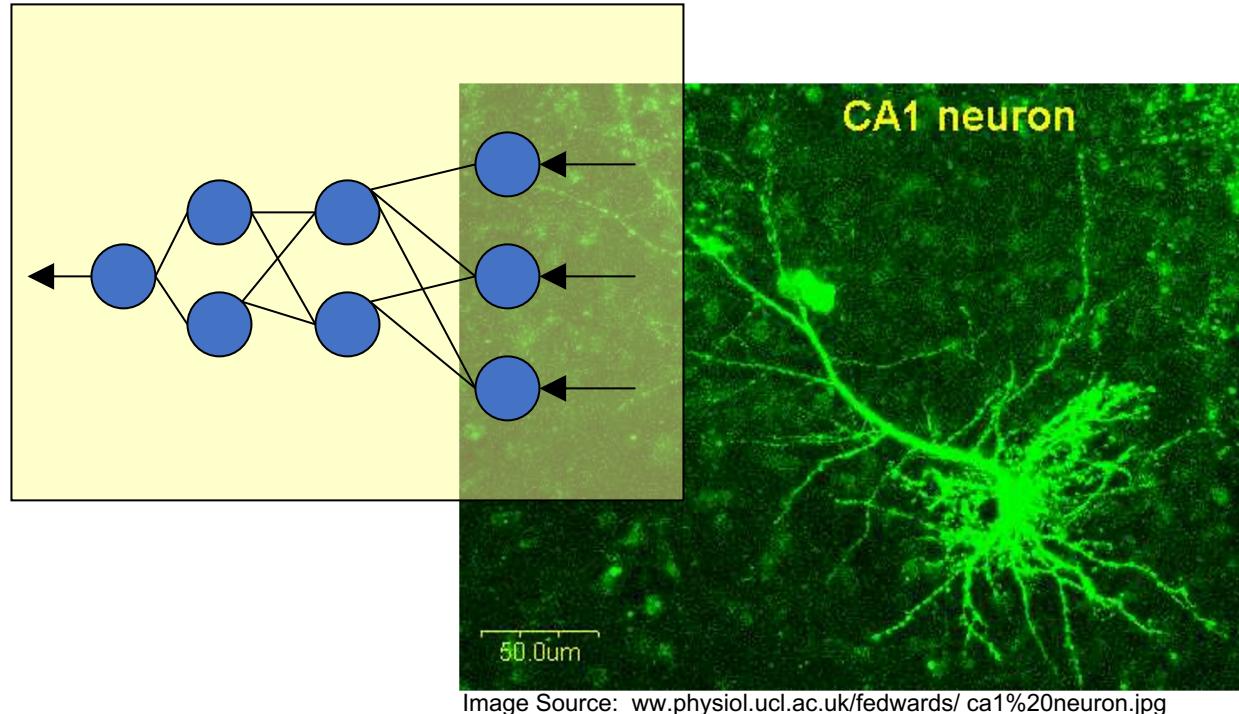


Outline

- Definisi Neural Network
- ANN – Feed-forward Network
- Ilustrasi

Definisi

Pemodelan Jaringan Syaraf Tiruan (Artificial Neural Network)



Jaringan Syaraf (Neural Network)

Suatu upaya pemodelan yang menirukan fungsi yang ada di otak manusia

Terdapat beberapa kelas model NN yang dibedakan berdasarkan:

- Tujuan pemodelan
Prediction, Classification, Clustering
- Struktur model
- Algoritma estimasi parameter model

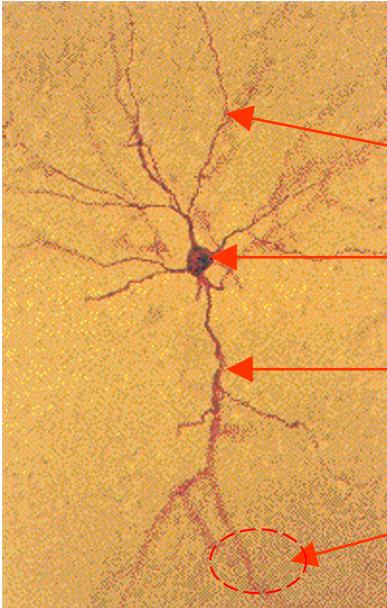
Materi ini akan fokus pada

Feed-forward Back-propagation Neural Network

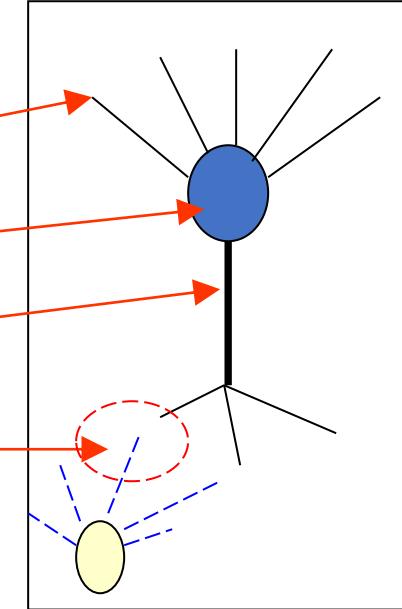
(digunakan untuk permasalahan Prediction and Classification)

A bit of biology . . .

Unit dengan fungsi terpenting dalam otak manusia – jaringan bernama– **NEURON**



Dendrit
Tubuh Sel
Axon
Sinapsis



Hippocampal Neurons

Source: heart.cbl.utoronto.ca/~berj/projects.html

Schematic

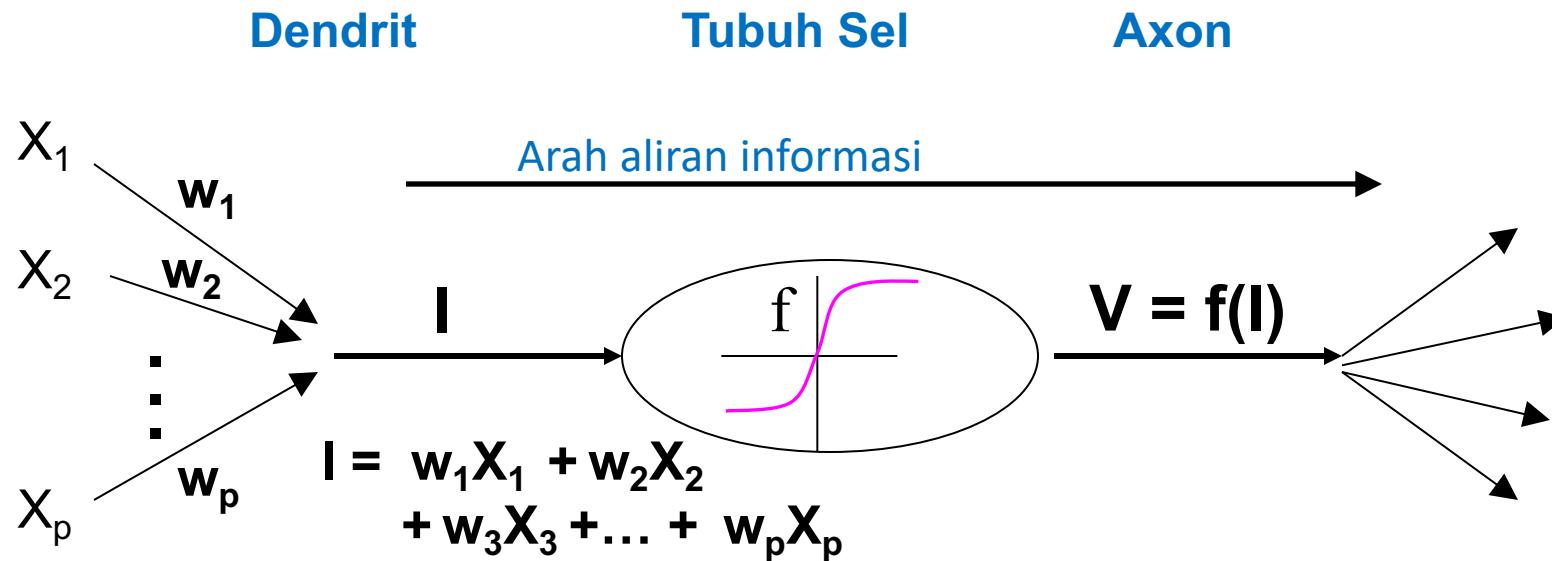
Dendrit – Menerima informasi

Tubuh Sel – Proses informasi

Axon – Membawa informasi yang telah diproses ke neuron lain

Sinapsis – Penghubung antara ujung Axon dan Dendrites neuron lain

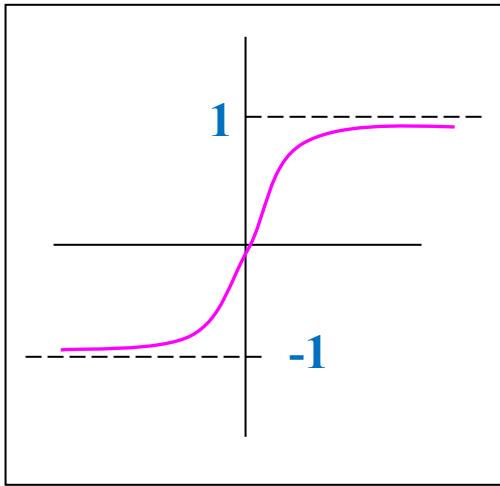
An Artificial Neuron



- Input X_1, X_2, \dots, X_p diterima dari neuron lain atau lingkungan
- Setiap input masuk ke dalam proses ini dengan bobot masing-masing
- Total input = jumlah terboboti semua input dari semua sumber
- Transfer function (Activation function) mengkonversi input ke output
- Output masuk ke neuron lain atau lingkungan

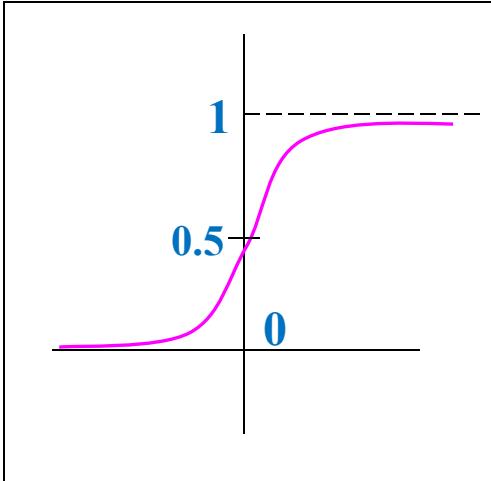
Transfer Functions

Banyak pilihan Transfer / Activation functions



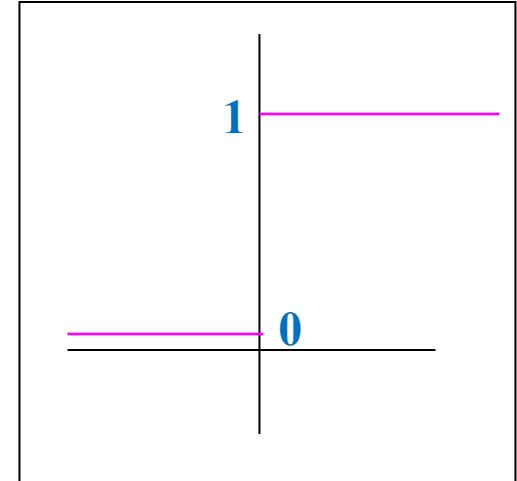
Tanh

$$f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$$



Logistic

$$f(x) = e^x / (1 + e^x)$$

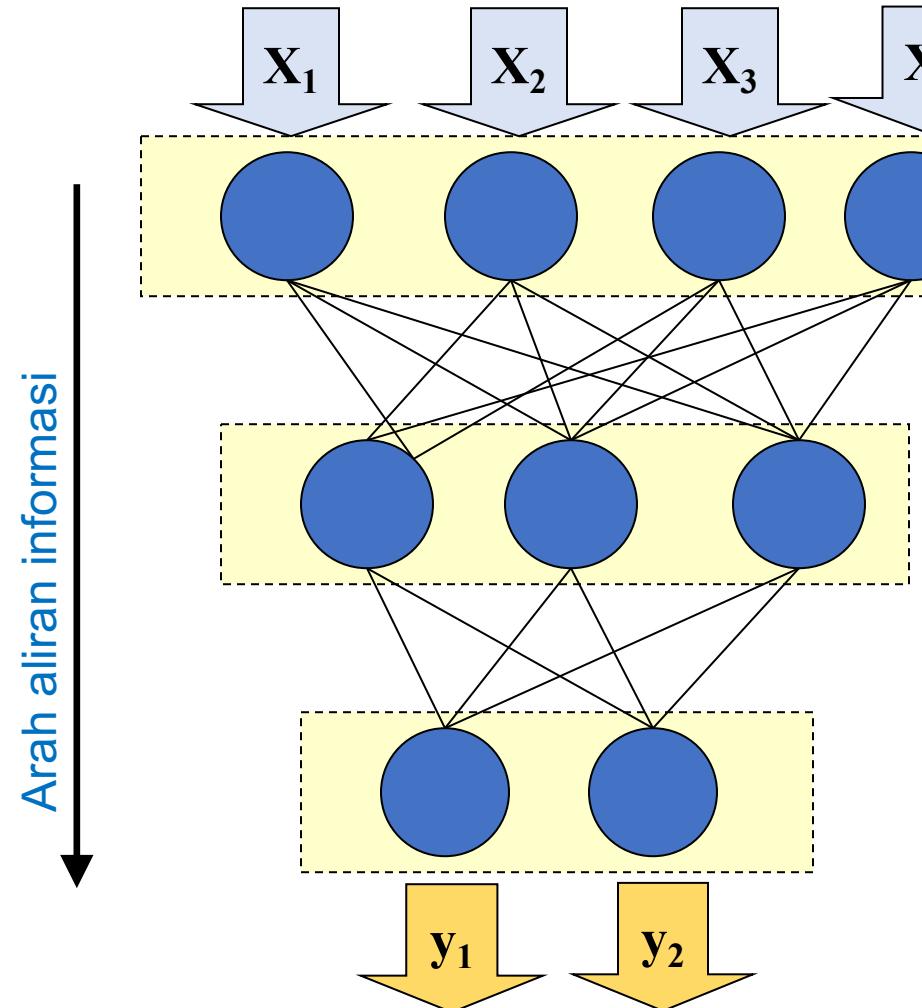


Threshold

$$f(x) = \begin{cases} 0; & \text{if } x < 0 \\ 1; & \text{if } x \geq 0 \end{cases}$$

ANN – Feed-forward Network

Sekelompok neuron pada level yang sama membentuk ‘Layer’



Input Layer

- Tiap neuron mendapat HANYA satu input, langsung dari luar

Hidden Layer

- Menghubungkan layer input dan output

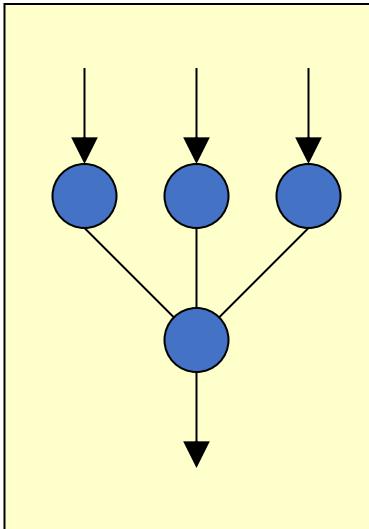
Output Layer

- Keluaran dari pengolahan informasi

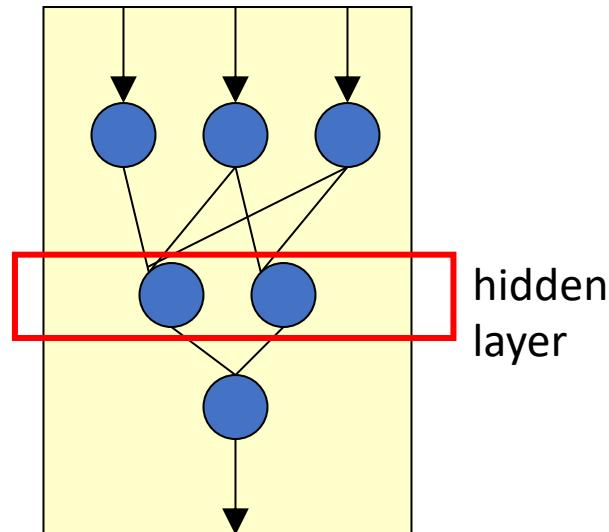
ANN – Feed-forward Network

Banyaknya hidden layer yang mungkin

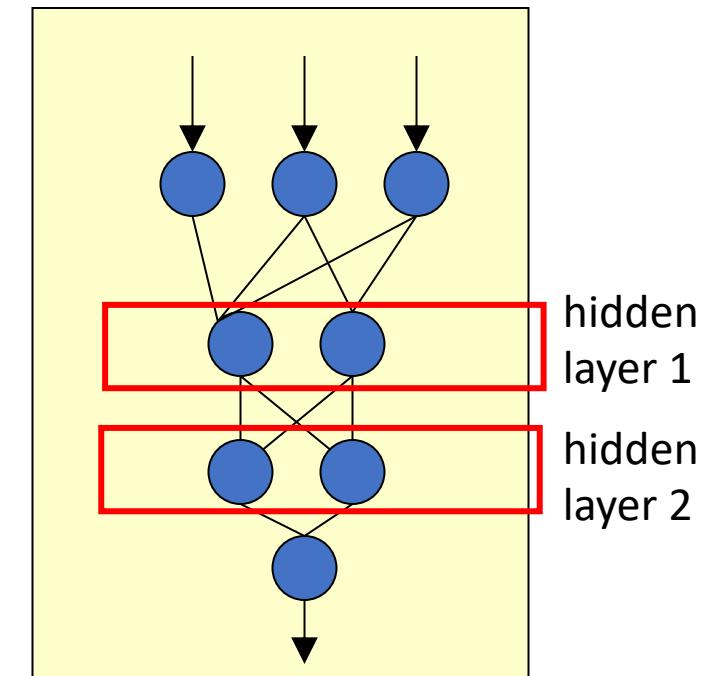
Tidak ada



Satu

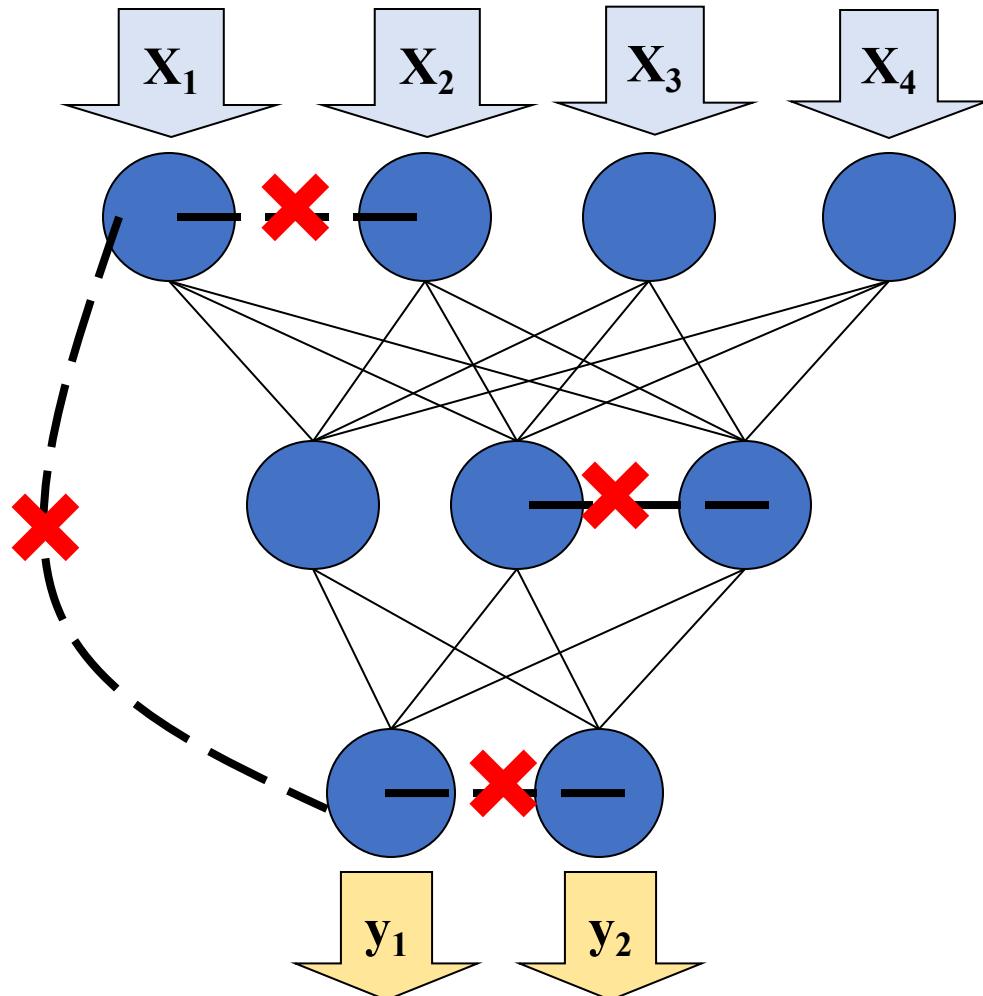


Lebih dari satu



ANN – Feed-forward Network

Beberapa hal yang perlu diperhatikan



- Neuron dalam satu layer TIDAK terhubung satu sama lain.
- Neuron dalam satu layer terhubung HANYA dengan layer BERIKUTNYA. (Feed-forward)
- Lompat layer TIDAK diperbolehkan

Model ANN

Komponen di dalam pemodelan

Input: $X_1 X_2 X_3$ Output: Y Model: $Y = f(X_1 X_2 X_3)$

Tidak seperti model regresi, persamaan matematika untuk ANN terlalu kompleks untuk dituliskan

Namun, ANN dicirikan dengan

- # Neuron input
- # Hidden Layer
- # Neuron dalam setiap Hidden Layer
- # Neuron output
- BOBOT untuk semua koneksi

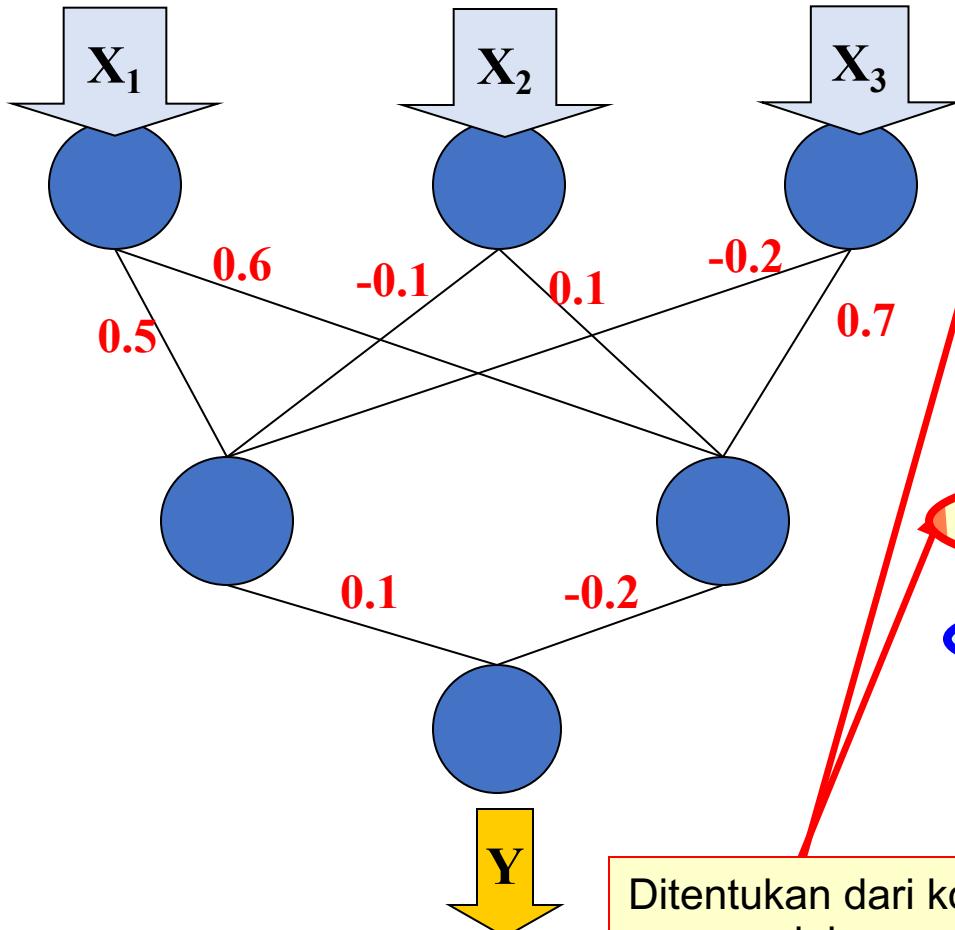
'Fitting' model ANN = menentukan nilai untuk semua parameter tersebut

Ilustrasi Model ANN

Input: $X_1 X_2 X_3$

Output: Y

Model: $Y = f(X_1 X_2 X_3)$



Parameter	Ilustrasi
# Neuron Input	3
# Hidden Layers	1
# Ukuran Hidden Layer	2
# Neuron Output	1
Bobot	Estimasi

Ditentukan dari konteks
permasalahan
Input Nrns = # of X's
Output Nrns = # of Y's

Free parameters

Prediksi menggunakan Model ANN

Input: $X_1 X_2 X_3$

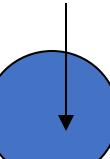
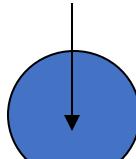
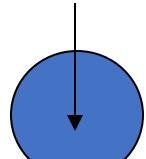
Output: Y

Model: $Y = f(X_1 X_2 X_3)$

$X_1 = 1$

$X_2 = -1$

$X_3 = 2$



0.5

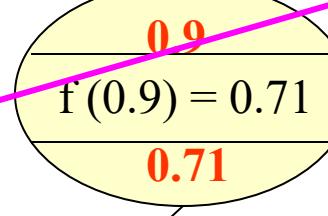
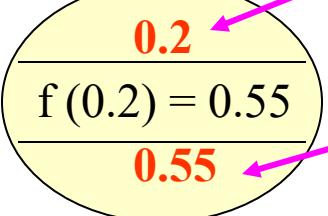
0.6

-0.1

0.1

-0.2

0.7



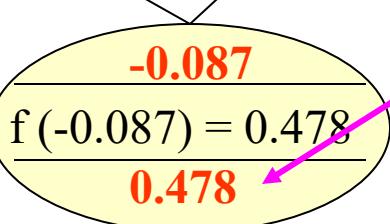
0.55

0.9

0.71

0.1

-0.2



$$0.2 = 0.5 * 1 - 0.1 * (-1) - 0.2 * 2$$

$$f(x) = e^x / (1 + e^x)$$
$$f(0.2) = e^{0.2} / (1 + e^{0.2}) = 0.55$$

Prediksi $Y = 0.478$

Misalkan Y Aktual = 2
maka
Prediction Error = $(2 - 0.478) = 1.522$

Membangun Model ANN

Bagaimana membangun model ANN?

Input: $X_1 X_2 X_3$

Output: Y

Model: $Y = f(X_1 X_2 X_3)$

Neuron Input = # Inputs = **3**

Neuron Output = # Outputs = **1**

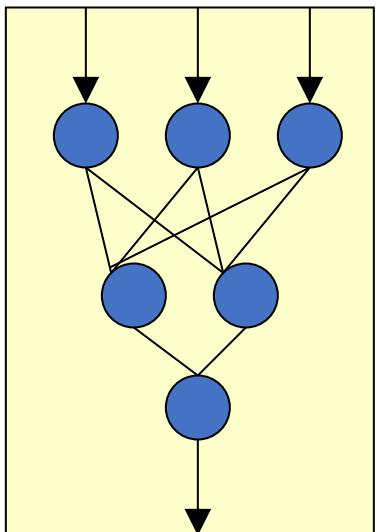
Hidden Layer = ???

Coba **1**

Neuron dalam Hidden Layer = ???

Coba **2**

Tidak ada strategi baku, dilakukan secara trial and error



Arsitektur model telah terdefinisi ... Bagaimana bobotnya???

Ilustrasi di samping, terdapat 8 bobot.

$$\underline{W} = (W_1, W_2, \dots, W_8)$$

Training Data: $(Y_i, X_{1i}, X_{2i}, X_{3i}) \quad i = 1, 2, \dots, n$

Untuk bobot W tertentu, diperoleh prediksi Y

$$(V_1, V_2, \dots, V_n)$$

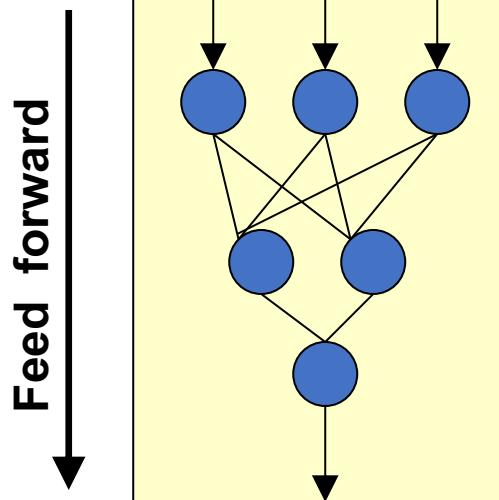
Prediksi ini adalah *fungsi* dari W.

Bobot W dipilih sedemikian rupa sehingga total prediction error **E** minimum

$$E = \sum (Y_i - V_i)^2$$

Training the Model

How to train the Model ?

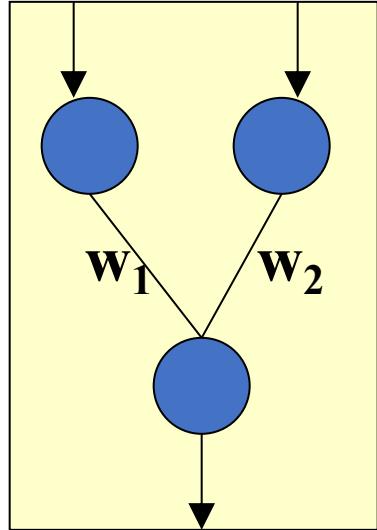


$$E = \sum (Y_i - V_i)^2$$

- Mulai dengan bobot awal.
- Feed forward pengamatan pertama melalui NN
 $X_1 \rightarrow \text{Network} \rightarrow V_1 ; \text{Error} = (Y_1 - V_1)$
- Sesuaikan bobot sehingga error ini mengecil
(network mengepas dengan baik pengamatan pertama)
- Feed forward pengamatan kedua.
Sesuaikan bobot agar NN mengepas dengan baik pengamatan kedua
- Ulang terus proses ini hingga pengamatan terakhir
- Sampai di sini, training PUTARAN pertama selesai
- Lakukan beberapa putaran hingga total prediction error E kecil.

Interpretasi geometrik dari penyesuaian bobot

Perhatikan NN dengan 2 input dan 1 output, tanpa hidden layer. NN ini memerlukan dua bobot yang harus ditentukan nilainya.



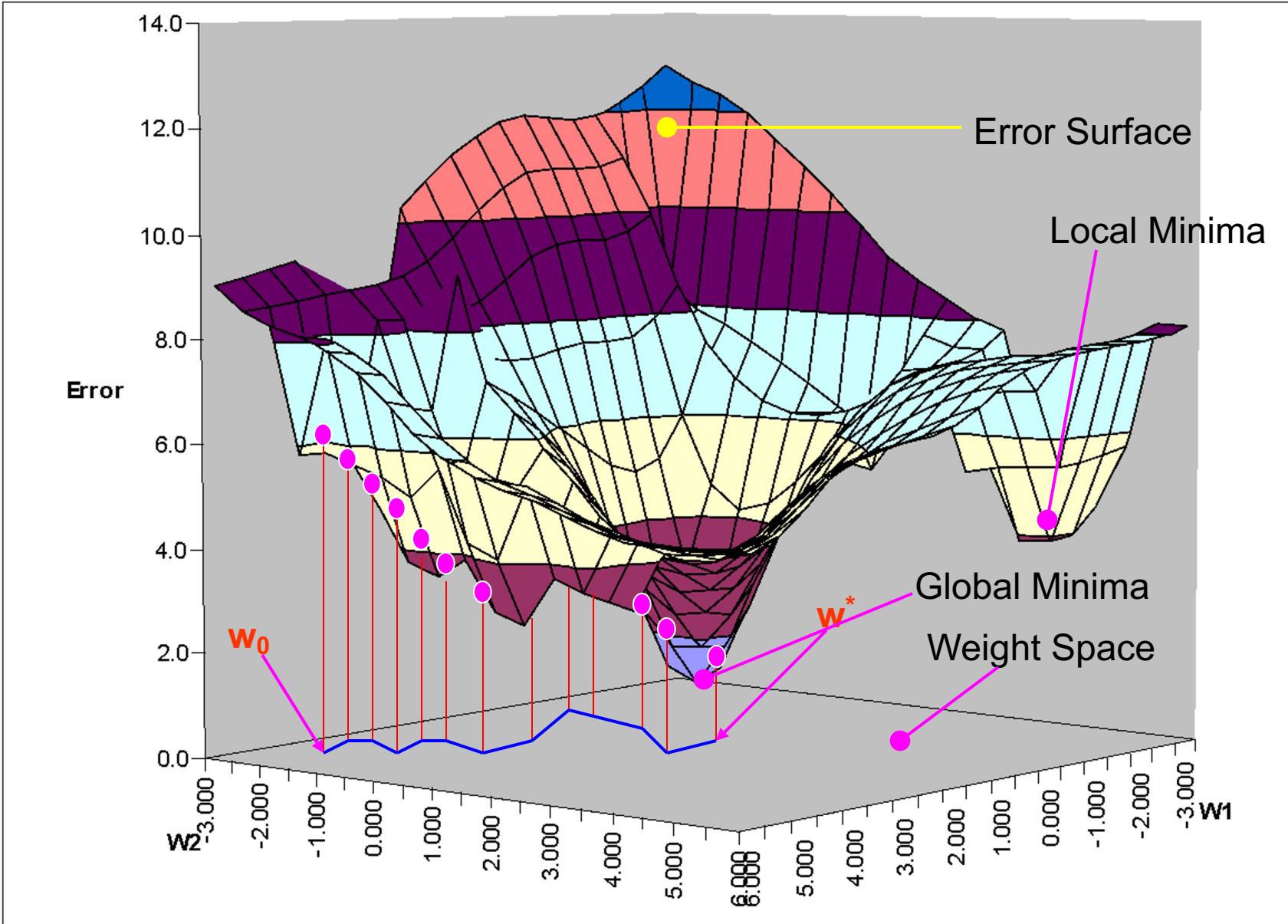
$$E(w_1, w_2) = \sum [Y_i - V_i(w_1, w_2)]^2$$

- Pasangan (w_1, w_2) adalah sebuah titik pada bidang 2-D.
- Untuk setiap titik dapat dihitung nilai E .
- Buat Plot E vs (w_1, w_2) pada ruang 3-D - '**Error Surface**'
- Tujuan kita adalah menentukan pasangan bobot dengan nilai E minimum
- Ini berarti kita berupaya mencari pasangan bobot dengan tinggi error surface minimum.

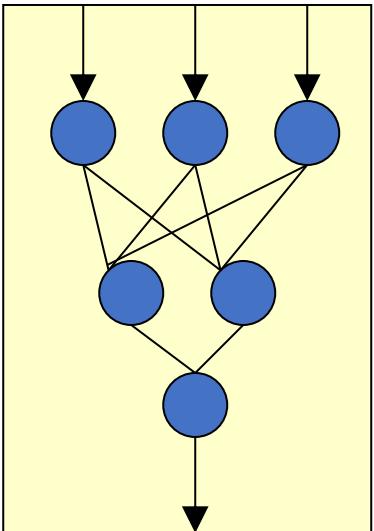
Gradient Descent Algorithm

- Mulai dengan sembarang titik (w_1, w_2)
- Pindah pada titik baru (w'_1, w'_2) dengan tinggi error surface lebih rendah.
- Terus cari titik baru hingga (w^*_1, w^*_2) , di mana error minimum.

Crawling the Error Surface



Training Algorithm



$$E = \sum (Y_i - V_i)^2$$

Tentukan Network architecture
(# Hidden layer, # Neuron pada setiap Hidden Layer)

Tentukan Learning parameter and Momentum

Initialize NN dengan sembarang bobot

Do till Convergence criterion is met

For $I = 1$ to # Training Data points

 Feed forward pengamatan ke- i
 Hitung prediction error pada pengamatan ke- i

 Back propagate error dan sesuaikan nilai bobot

Next I

Check for Convergence

End Do

Back propagation

- Setiap bobot ikut menyumbang kesalahan (Shares the Blame) pada prediction error bersama-sama dengan bobot lainnya.
- Algoritma Back Propagation akan memutuskan bagaimana mendistribusikan kesalahan ini pada semua bobot dan sekaligus menyesuaikan nilainya.
- Porsi kesalahan yang kecil berujung pada penyesuaian kecil pada bobot.
- Porsi kesalahan yang besar membawa efek pada penyesuaian nilai bobot yang juga besar.

Convergence Criterion

Kapan diputuskan training NN selesai?

Idealnya – ketika tercapai **global minima** pada error surface

Bagaimana kita tahu titik ini telah tercapai? Kita tidak pernah tahu ...

Suggestion:

1. Stop jika penurunan total prediction error (dari putaran terakhir) **kecil**.
2. Stop jika keseluruhan perubahan bobot (dari putaran terakhir) **kecil**.

Drawback:

Error terus menurun. Pada titik ini kita mendapatkan model **yang sangat cocok dengan data training**.

NAMUN ... Model NN memiliki performa yang tidak bagus (**poor generalizing power**) pada data lain
(unseen data)

Fenomena ini dinamakan - **Over fitting** pada data training

Convergence Criterion

Modified Suggestion:

Partisi data menjadi **Training set** dan **Validation set**

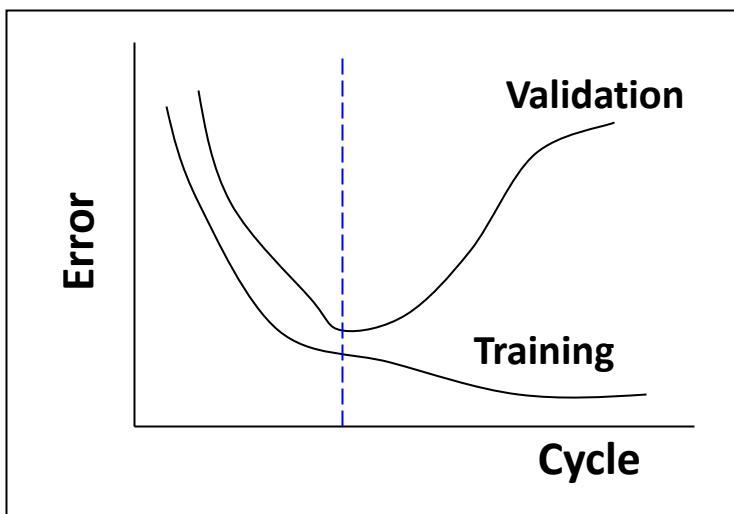
Gunakan Training set – membangun model

Validation set – Uji performa model pada data lain

Biasanya dengan semakin banyaknya putaran training

Error pada Training set terus menurun.

Error pada Validation set awalnya menurun kemudian meningkat.



Stop training ketika error pada Validation set mulai naik

Pemilihan Parameter Training

Learning Parameter dan Momentum

- harus ditentukan oleh pengguna, dengan nilainya antara 0 dan 1

Berapa nilai yang optimal bagi kedua parameter ini?

- Tidak ada strategi baku penentuan nilai optimal.
- Hanya saja, efek dari kesalahan penentuan nilai kedua parameter ini dapat ditelusuri.

Learning Parameter

Terlalu besar – Perubahan titik bobot setiap putaran sangat besar – risiko global minima terlewatkan.

Terlalu kecil –

- Memakan waktu lama untuk konvergen pada global minima
- Sekali terperangkap pada local minima, sulit untuk keluar.

Suggestion

Trial and Error – Cobakan berbagai nilai Learning Parameter dan Momentum dan perhatikan nilai mana yang berujung pada prediction error minimum

Wrap Up

- Artificial Neural network (ANN)
 - Pemodelan yang terinspirasi dari prinsip kerja jaringan syaraf
- Digunakan untuk berbagai tujuan pemodelan – Prediction, Classification, Clustering, ..
- Salah satu metode dalam ANN – Feed forward Back propagation networks
 - ❖ Dibentuk dari beberapa layer – Input, hidden, Output
 - ❖ Tiap layer adalah sekumpulan artificial Neurons
 - ❖ Neuron dalam satu layer terhubung dengan neuron pada layer berikutnya
 - ❖ Hubungan antar neuron memiliki bobot tertentu
- Fitting model ANN pada dasarnya adalah mencari nilai optimal bobot ini.
- Berdasarkan data training set – bobot diperoleh dengan algoritma Feed forward Back propagation, yang menerapkan metode Gradient Descent Method – yaitu suatu teknik yang populer digunakan untuk meminimumkan suatu fungsi.
- Arsitektur Network maupun parameter training ditentukan secara trial and error. Cobakan berbagai nilai yang mungkin dan pilih yang menghasilkan prediction error minimum.

Ilustrasi

Gunakan data iris di R

1. Lakukan ANN untuk Binary classification untuk Species = "setosa"
2. Lakukan ANN untuk Multiclass classification untuk Species
3. Lakukan ANN untuk Binary classification untuk Species = "setosa" dengan fungsi aktivasi:

$$\ln(1 + e^x)$$

```
library(neuralnet)

# Binary classification
nn <- neuralnet(Species == "setosa" ~ Petal.Length + Petal.Width, iris,
linear.output = FALSE)

print(nn)
plot(nn)

# Multiclass classification
nn2 <- neuralnet(Species ~ Petal.Length + Petal.Width, iris, linear.output = FALSE)
names(nn2)
nn2$act.fct

print(nn2)
plot(nn2)

# Custom activation function
softplus <- function(x) log(1 + exp(x))
nn3 <- neuralnet((Species == "setosa") ~ Petal.Length + Petal.Width, iris,
linear.output = FALSE, hidden = c(3, 2), act.fct = softplus)
print(nn3)
plot(nn3)
```

Tugas individu

- Suatu perusahaan perbank-kan meneliti 75 jenis skema pinjaman yang telah diberi rating oleh para customernya pada **data ann.csv** (data terlampir di newlms)
- Variabel yang digunakan ialah:
 - Besar pinjaman (dalam juta rupiah)
 - Lama pembayaran (dalam tahun)
 - Tambahan bunga yang ditetapkan (dalam %)
 - Pembayaran per bulan (dalam 10000)
 - Banyak cash back yang diterapkan pada skema tersebut
- Tujuan penelitian yang dilakukan ialah memprediksi rating skema pinjaman berdasarkan variabel-variabel tersebut
- Bantulah peneliti tersebut untuk memprediksi reating skema pinjaman dengan neural network, hitunglah RMSE prediksinya

Ketentuan Tugas:

- Kirimkan kode R beserta output dan interpretasinya pada file dengan format pdf
- Submit tugas individu pada newlms STA1382 – Teknik Pembelajaran Mesin Genap 2022-2023, pada Kuliah 5 – Neural Network, Tugas Individu 1
- Batas waktu pengiriman adalah **Hari Senin, tanggal 27 Februari 2023 jam 10:00 WIB**
- Komponen Penilaian:
 - Kecepatan pengiriman
 - Kesesuaian jawaban
 - Orisinalitas

Terima kasih 😊



Support Vector Machines

Kuliah 6 - STA1382 Teknik
Pembelajaran Mesin

Septian Rahardiantoro



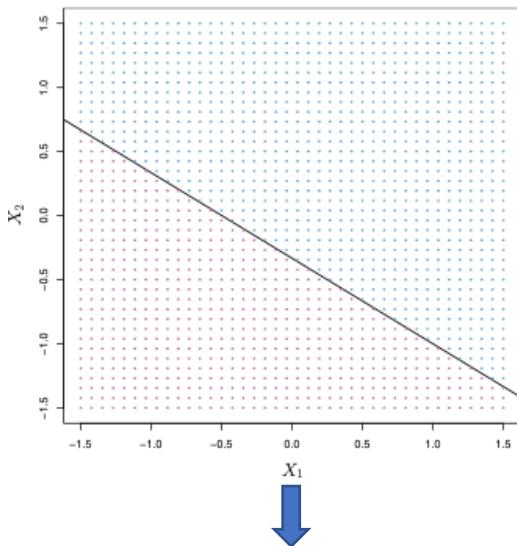
Outline

- Pengantar
- Hyperplane?
- Maximal Margin Classifier
- Support Vector Classifier
- Support Vector Machines

Pengantar

- *Support vector machine* (SVM) merupakan sebuah pendekatan untuk klasifikasi yang dikembangkan dalam komunitas ilmu komputer pada 1990-an dan semakin populer sejak saat itu.
- SVM telah terbukti berkinerja baik dalam berbagai pengaturan, dan sering dianggap sebagai salah satu pengklasifikasi "out of the box" terbaik.
- SVM adalah generalisasi dari pengklasifikasi sederhana dan intuitif yang disebut *marginal margin classifier* dan pengembangannya *support vector classifier*

Ide dasar dan konteks dalam SVM



hyperplane

marginal margin classifier pengklasifikasi sederhana dan intuitif



support vector classifier



support vector machines

pengembangan dari *marginal margin classifier* yang dapat diterapkan dalam rentang kasus yang lebih luas

Pengembangan lebih lanjut dari *support vector classifier* untuk mengakomodasi batas kelas non-linear.

Hyperplane?

- Dalam ruang berdimensi p , hyperplane adalah subruang bidang datar berdimensi $p - 1$
 - Misalkan dalam 2 dimensi, hyperplane adalah subruang datar satu dimensi → suatu garis lurus
 - Dalam 3 dimensi, hyperplane adalah subruang datar dua dimensi → suatu bidang datar
- Pada ruang berdimensi p , hyperplane didefinisikan sebagai:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

- untuk setiap $X = (X_1, X_2, \dots, X_p)^T$ ✓

- Maka untuk 2 dimensi, hyperplane → $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$ (garis lurus), untuk $X = (X_1, X_2)^T$

- Pada ruang berdimensi p , anggap:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \longrightarrow X \text{ jatuh pada sisi } \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \longrightarrow X \text{ jatuh pada sisi yang lainnya}$$

- Jadi kita bisa menganggap hyperplane membagi ruang berdimensi p menjadi dua bagian.

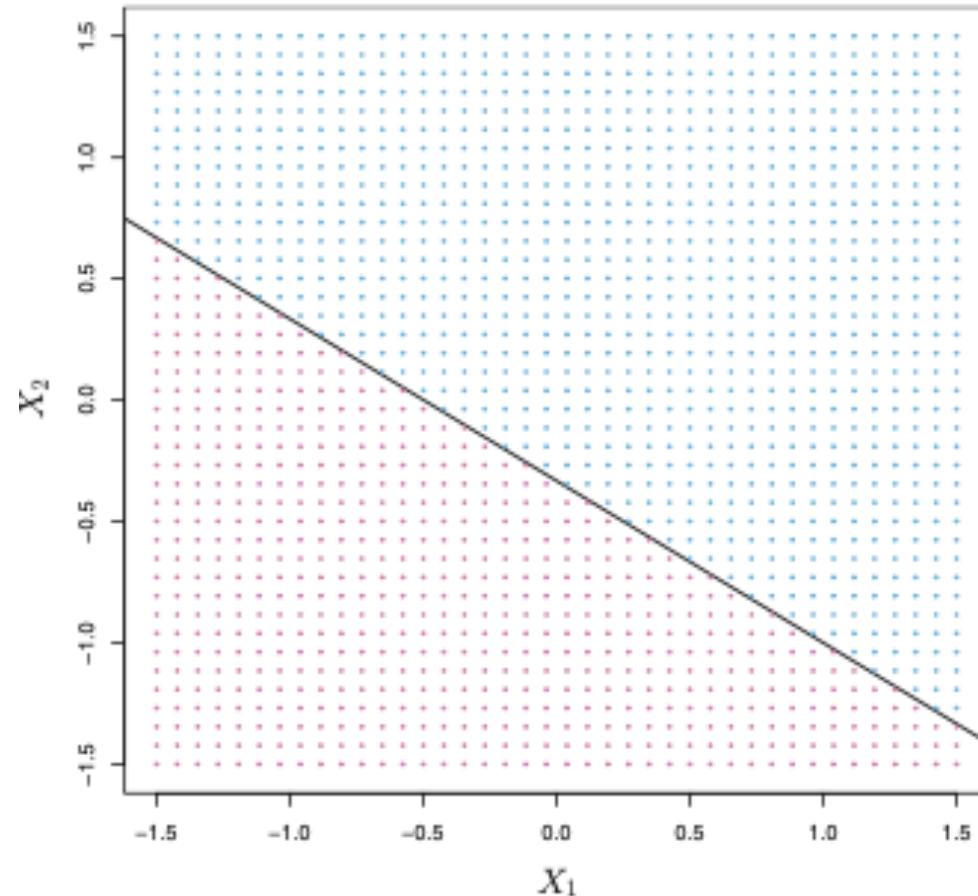
- Seseorang dapat dengan mudah menentukan di sisi mana dari hyperplane suatu titik terletak dengan hanya menghitung tanda $\underline{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p}$

$\vec{x} = (x_1, x_2, \dots, x_p)^T$

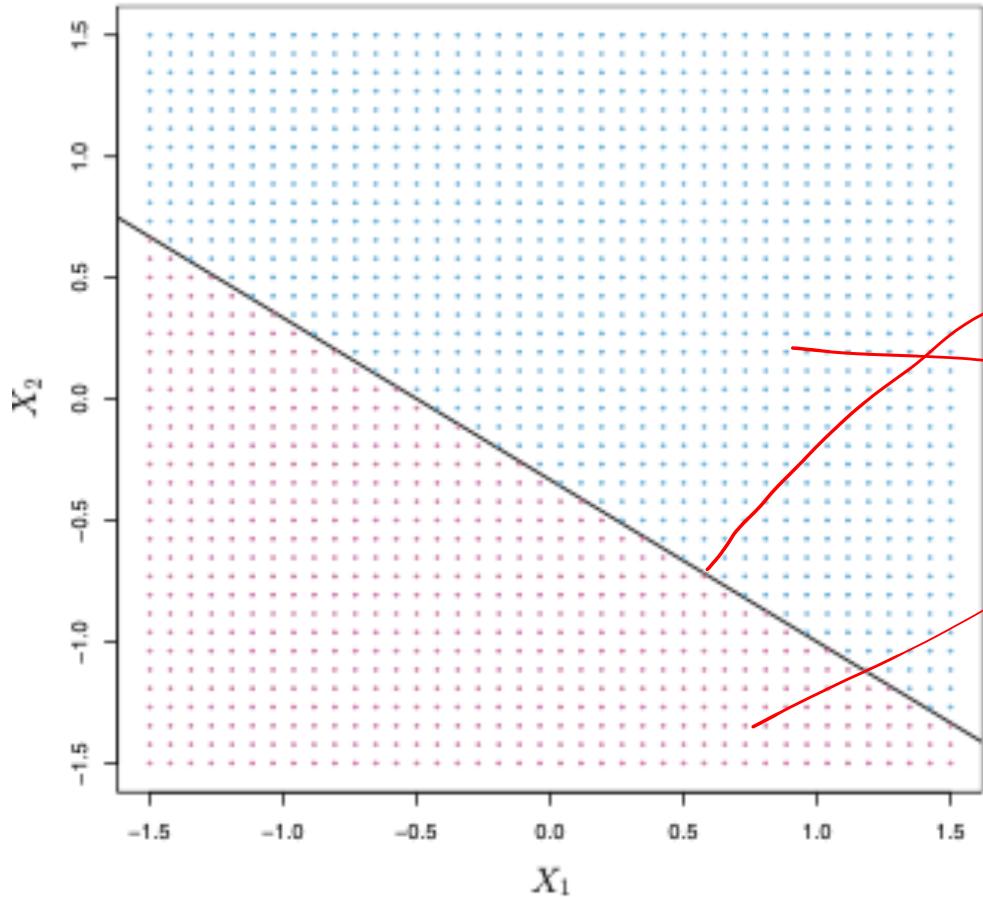
$\rightarrow > 0$

$\rightarrow < 0$

$\vec{x} = (3, -1, 5)$



Contoh sebuah hyperplane dalam ruang dua dimensi



Contoh sebuah hyperplane dalam ruang dua dimensi

$$\text{Hyperplane } 1 + 2X_1 + 3X_2 = 0$$

Daerah biru adalah himpunan titik dimana $1 + 2X_1 + 3X_2 > 0$

Daerah ungu adalah himpunan titik dimana $1 + 2X_1 + 3X_2 < 0$

$$\begin{aligned}
 X_1 &= 5 \\
 X_2 &= -1
 \end{aligned}
 \rightarrow 1 + 2 \cdot 5 + 3 \cdot (-1) \\
 &= 1 + 10 + -3 \\
 &= 8 > 0 \rightarrow \text{biru.}$$

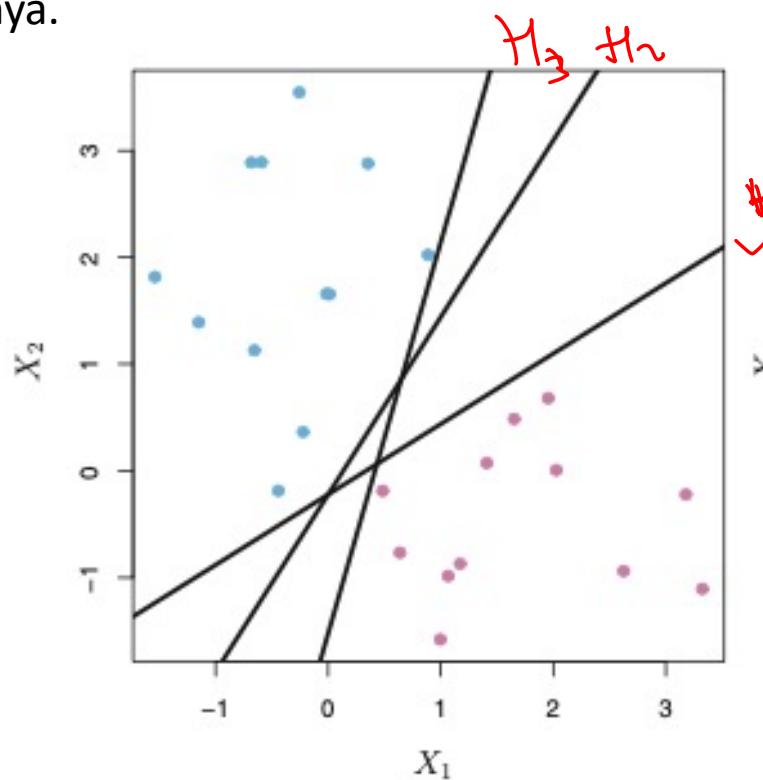
Klasifikasi Menggunakan Pemisahan Hyperplane

- Sekarang misalkan diketahui matriks data training X berdimensi $n \times p$, yang terdiri dari n observasi dalam ruang dimensi- p

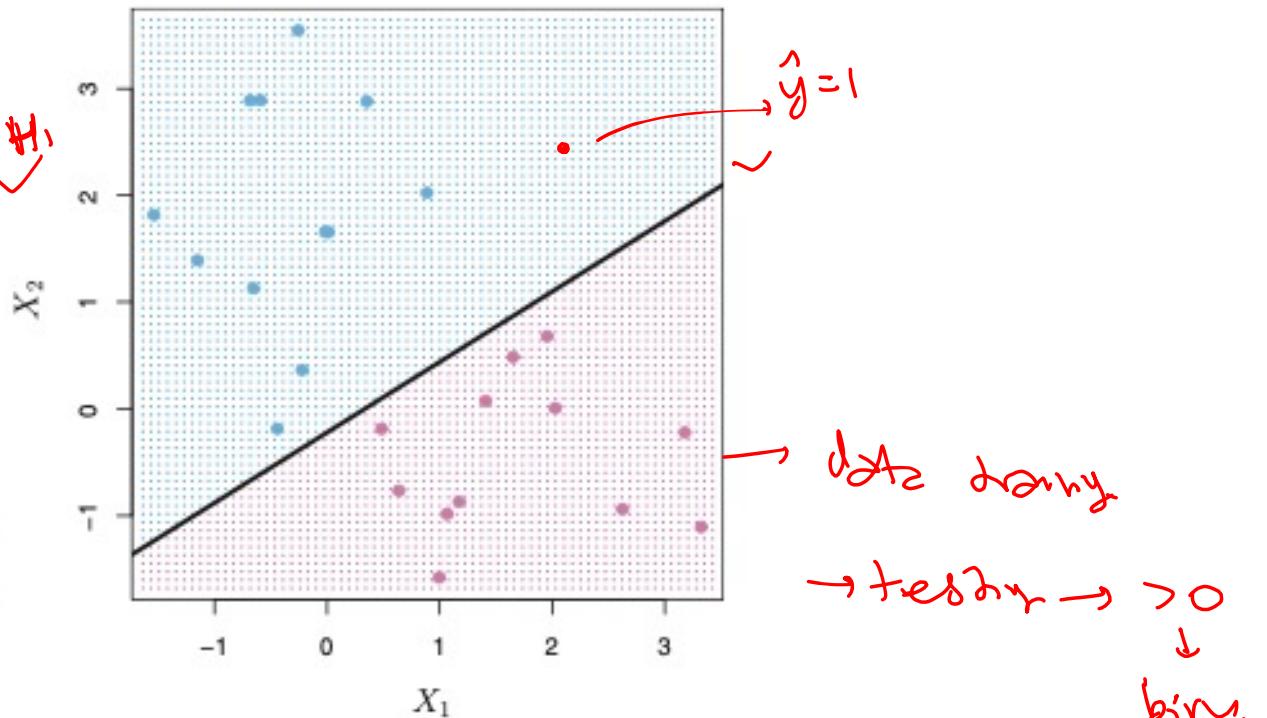
$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \dots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix} \rightarrow \text{pada kelas } y_1, \dots, y_n \in \{-1, 1\}.$$

- Tujuannya adalah mengembangkan pengklasifikasi berdasarkan data training yang akan mengklasifikasikan pengamatan data testing dengan benar menggunakan pengukuran fiturnya. $\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$
- Sekarang kita akan melihat pendekatan baru yang didasarkan pada konsep memisahkan hyperplane.

- Misalkan dimungkinkan untuk membuat hyperplane yang memisahkan observasi pelatihan dengan sempurna sesuai dengan label kelasnya.



Ada dua kelas pengamatan, ditunjukkan dengan warna biru (1) dan ungu (-1), yang masing-masing memiliki pengukuran pada dua variabel. Tiga hyperplanes yang memisahkan, dari banyak kemungkinan, ditampilkan dalam warna hitam.



Kanan: Sebuah hyperplane pemisah ditampilkan dalam warna hitam. Kisi biru dan ungu menunjukkan aturan keputusan yang dibuat oleh pengklasifikasi berdasarkan hyperplane pemisah ini: observasi uji yang termasuk dalam bagian biru akan diduga ke kelas biru, dan observasi uji yang termasuk dalam bagian ungu akan diduga ke kelas ungu.

- Misalkan terdapat 2 kelas pengamatan: kelas warna biru sebagai $y_i = 1$ dan kelas warna ungu sebagai $y_i = -1$

maka hyperplane pemisah dapat didefinisikan sebagai

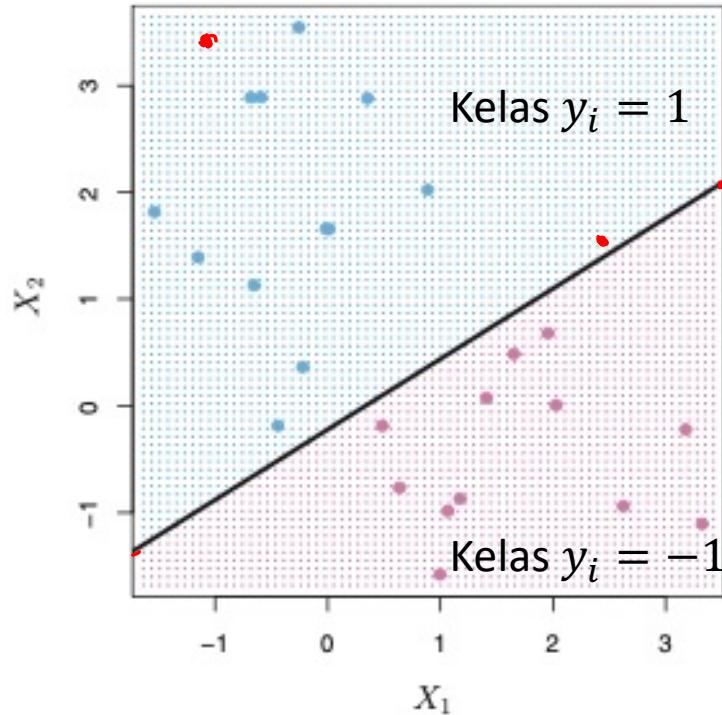
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 \text{ jika } y_i = 1 \quad \checkmark$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 \text{ jika } y_i = -1 \quad \checkmark$$

- Secara ekuivalen, hyperplane pemisah memiliki properti

$$y_i(\underbrace{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}_{> 0}) > 0 \text{ untuk setiap } i = 1, 2, 3, \dots, n$$

- Jika terdapat hyperplane pemisah, kita dapat menggunakannya untuk membuat classifier yang sangat alami: observasi data testing diberi kelas tergantung pada sisi mana hyperplane itu berada. ✓



Misalkan dengan pengamatan data testing $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_p^*)$

Maka \mathbf{x}^* diklasifikasikan berdasarkan tanda dari

$$f(\mathbf{x}^*) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$$

Jika $f(\mathbf{x}^*) > 0$, maka \mathbf{x}^* diduga masuk ke kelas 1

Jika $f(\mathbf{x}^*) < 0$, maka \mathbf{x}^* diduga masuk ke kelas -1

$$y = 1$$

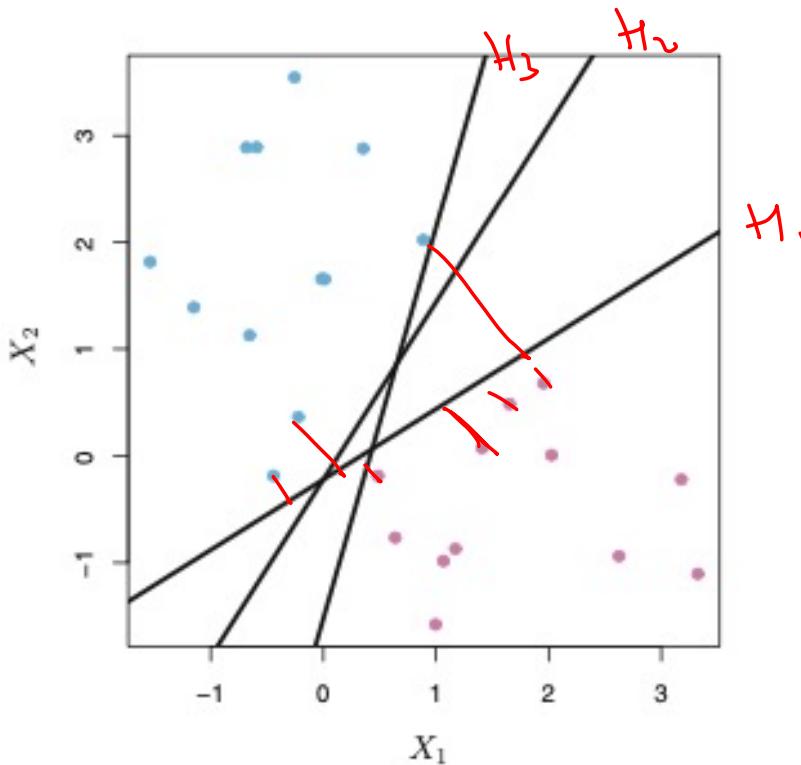
$$y = -1$$

Perhatikan, kita juga dapat melihat magnitude (ukuran) dari $f(\mathbf{x}^*)$

- Jika $f(\mathbf{x}^*)$ jauh dari 0 → \mathbf{x}^* berada jauh dari hyperplane → jadi kita bisa percaya diri tentang kelas untuk \mathbf{x}^*
- Jika $f(\mathbf{x}^*)$ mendekati 0 → \mathbf{x}^* berada dekat atau disekitar hyperplane → jadi kurang yakin tentang kelas untuk \mathbf{x}^*

Ide inilah yang nantinya diterapkan pada konsep maximal margin classifier

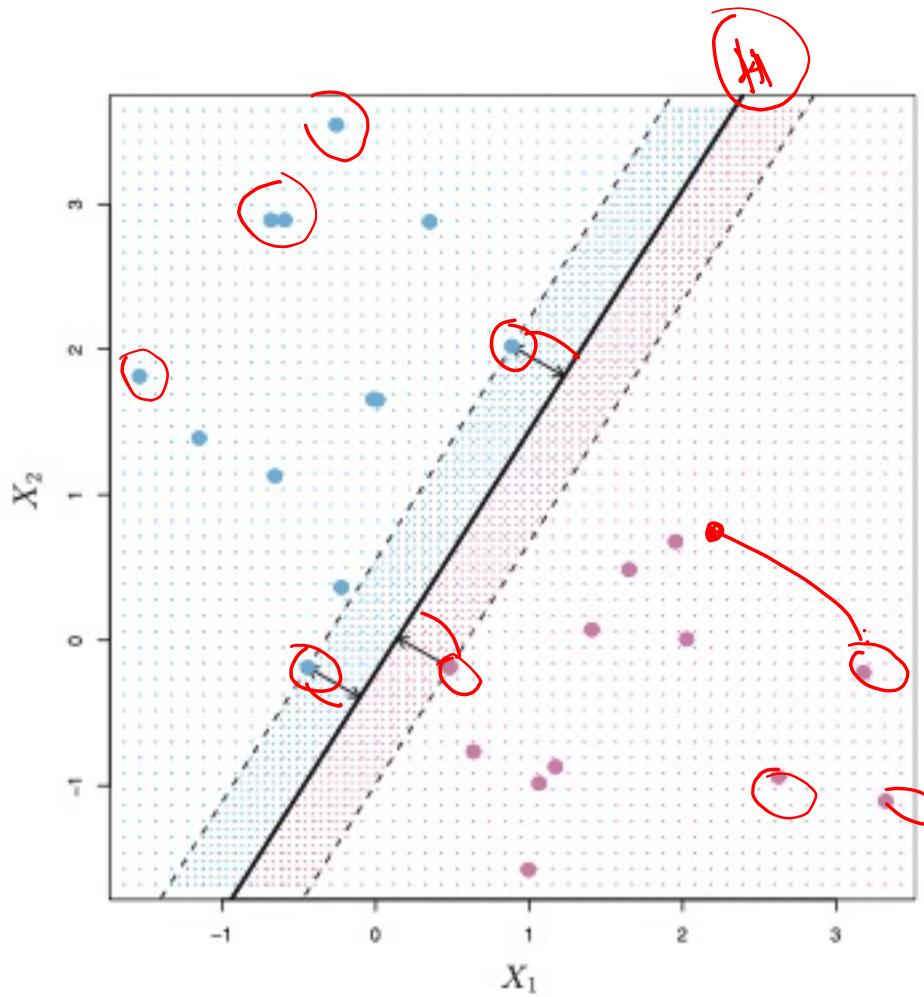
Maximal Margin Classifier



Tiga kemungkinan hyperplanes pemisah

- Untuk membuat pengklasifikasi berdasarkan hyperplane pemisah, kita harus memiliki cara yang masuk akal untuk memutuskan hyperplane pemisah mana yang akan digunakan.
- *Maximal margin hyperplane* (atau *optimal separating hyperplane*) merupakan hyperplane pemisah yang terjauh dari pengamatan data training.
- Yaitu, kita dapat menghitung jarak (tegak lurus) dari setiap observasi data training ke hyperplane pemisah yang diberikan; jarak terkecil adalah jarak minimal dari pengamatan ke hyperplane, dan dikenal sebagai margin.

- *Maximal margin hyperplane* adalah hyperplane pemisah yang memiliki margin terbesar—artinya, hyperplane tersebut memiliki jarak minimum terjauh ke pengamatan data training.
- Kemudian dapat diklasifikasikan pengamatan data testing berdasarkan sisi mana dari *maximal margin hyperplane* itu terletak. → Ini dikenal sebagai ***maximal margin classifier***.
- Harapannya, pengklasifikasi yang memiliki margin besar pada data training juga akan memiliki margin besar pada data testing, dan karenanya akan mengklasifikasikan pengamatan data testing dengan benar.
- Meskipun *maximal margin classifier* sering berhasil, hal ini juga dapat menyebabkan overfitting ketika p besar.
- Jika $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ adalah koefisien *maximal margin hyperplane*, maka *maximal margin classifier* mengklasifikasikan pengamatan data testing x^* berdasarkan tanda $f(x^*) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$



Ada dua kelas pengamatan, ditunjukkan dengan warna biru dan ungu.

- *Maximal margin hyperplane* ditampilkan sebagai garis solid berwarna hitam.
- Margin adalah jarak dari garis solid ke salah satu garis putus-putus.
- Dua titik biru dan titik ungu yang terletak pada garis putus-putus adalah vektor pendukung (*support vectors*), dan jarak dari titik tersebut ke margin ditunjukkan dengan panah.
- Grid ungu dan biru menunjukkan aturan keputusan yang dibuat oleh pengklasifikasi berdasarkan hyperplane pemisah ini.
- *Maximal margin hyperplane* mewakili garis tengah dari "lempengan" terluas yang dapat kita sisipkan di antara kedua kelas.
- Menariknya, *maximal margin hyperplane* bergantung langsung pada *support vectors*, tetapi tidak pada pengamatan lain: pergerakan ke salah satu pengamatan lain tidak akan mempengaruhi hyperplane pemisah, asalkan pergerakan pengamatan tidak menyebabkannya melintasi batas yang ditetapkan oleh margin.

Konstruksi *Maximal Margin Classifier*

- Secara singkat, *maximal margin hyperplane* adalah solusi untuk masalah optimisasi

$$\underset{\beta_0, \beta_1, \dots, \beta_p}{\text{maximize}} M$$

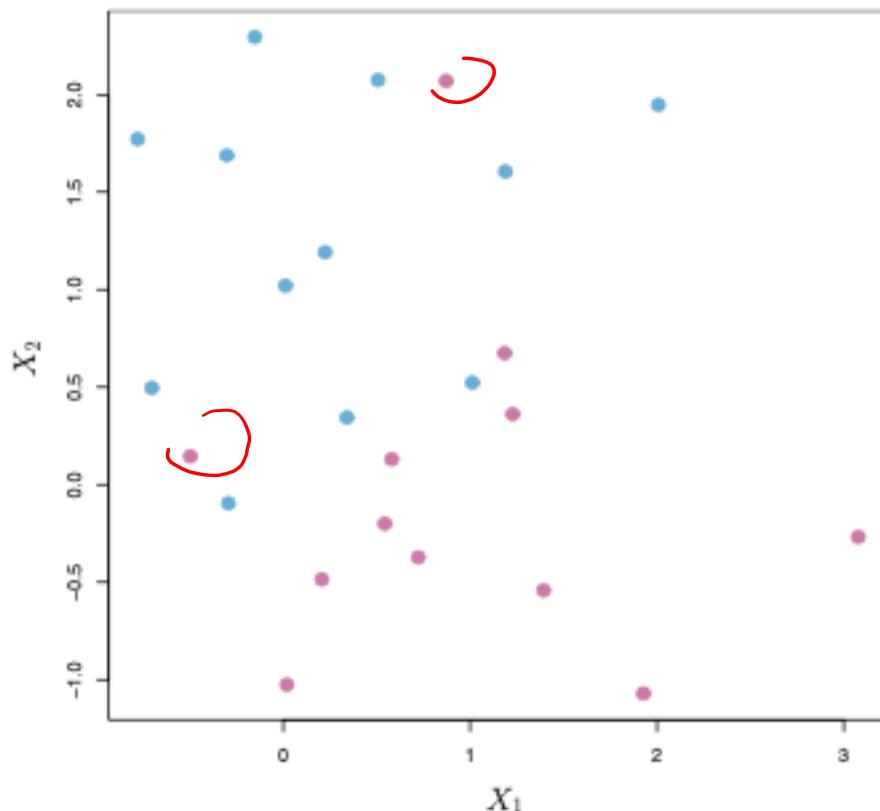
$$\text{subject to } \sum_{j=1}^p \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \dots, n.$$

Dua kendala ini memastikan bahwa setiap pengamatan berada di sisi yang benar dari hyperplane dan setidaknya berjarak M dari hyperplane.

Oleh karena itu, M merepresentasikan margin hyperplane kita, dan masalah optimisasi memilih $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ untuk memaksimalkan M . Ini persis definisi *maximal margin hyperplane!* ✓

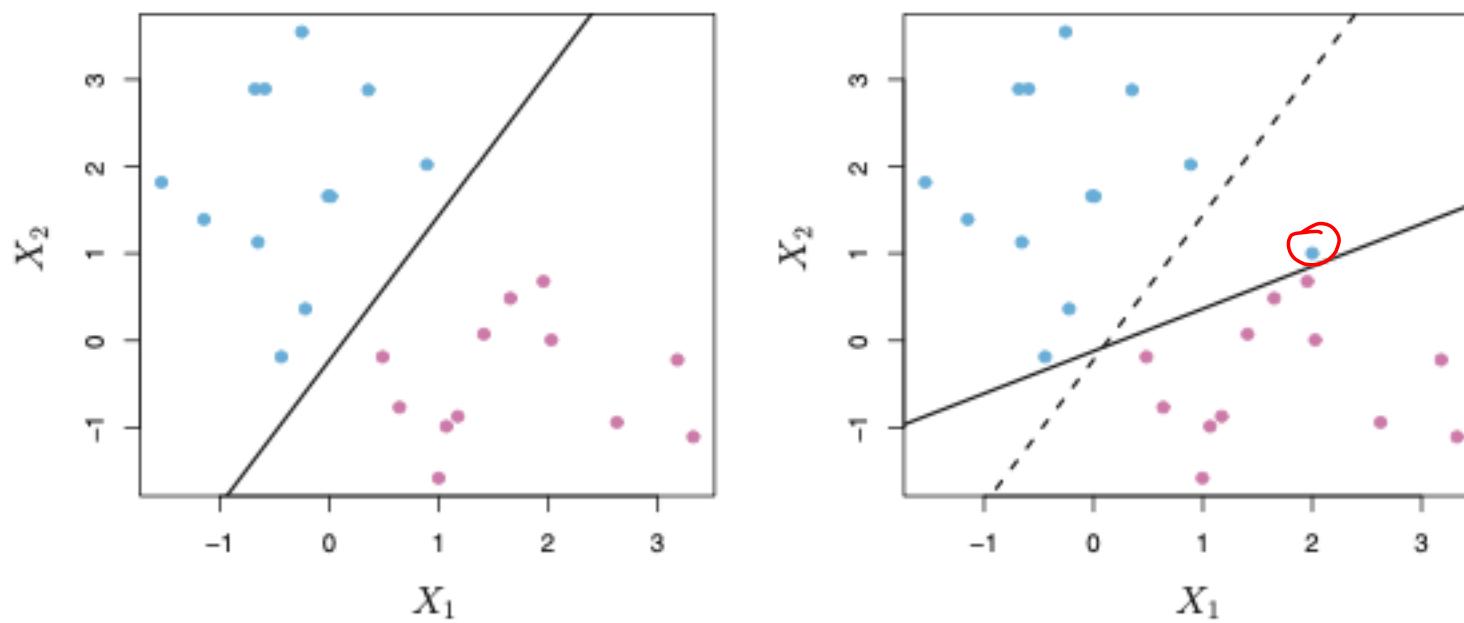
Kasus yang Tidak Dapat Dipisahkan



Ada dua kelas pengamatan, ditunjukkan dengan warna biru dan ungu. Dalam hal ini, kedua kelas tidak dapat dipisahkan oleh hyperplane, sehingga *maximal margin classifier* tidak dapat digunakan.

- *Maximal margin classifier* adalah cara yang sangat alami untuk melakukan klasifikasi, jika terdapat hyperplane pemisah.
- Namun, seperti pada kasus di samping, tidak ada hyperplane pemisah, sehingga tidak ada *maximal margin classifier*.
- Generalisasi dari *maximal margin classifier* ke kasus yang tidak dapat dipisahkan dikenal sebagai *support vector classifier*.

Support Vector Classifier

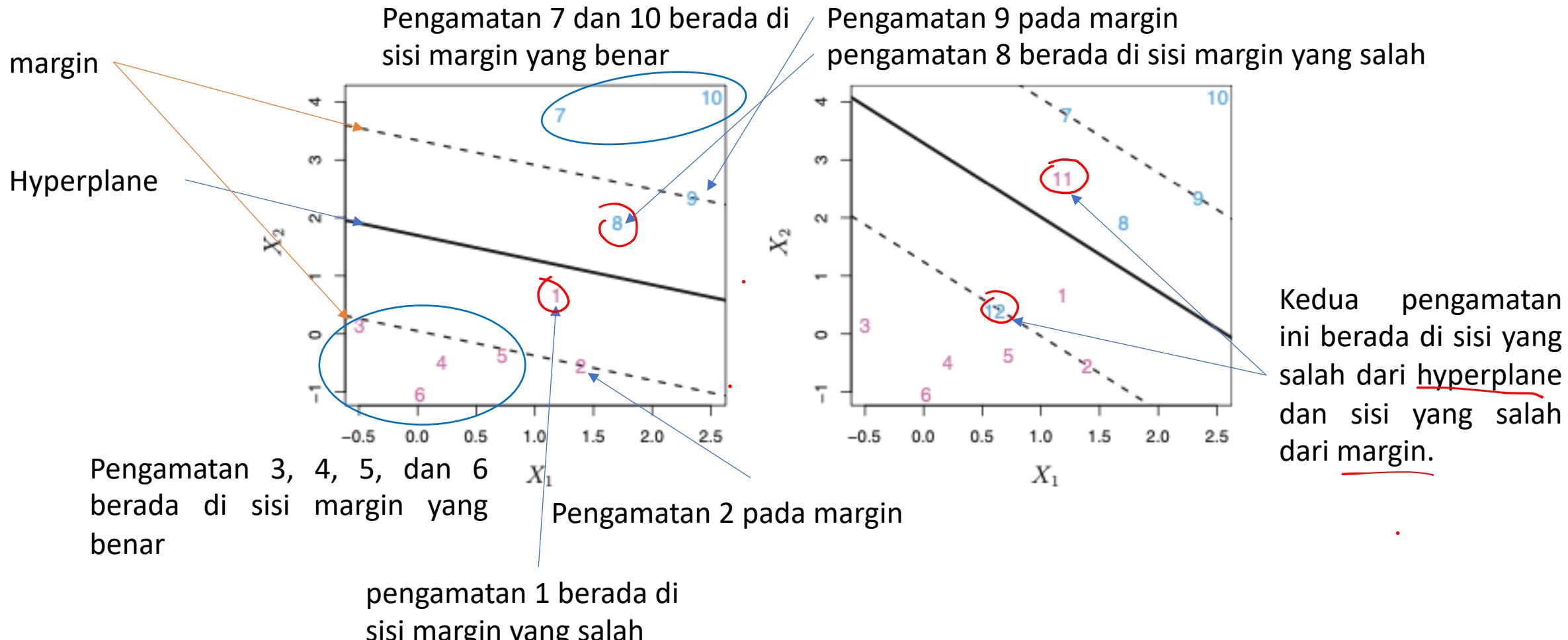


- Kiri: Dua kelas pengamatan ditampilkan dalam warna biru dan ungu, bersama dengan *maximal margin hyperplane*.
- Kanan: Pengamatan biru tambahan telah ditambahkan, mengarah ke pergeseran dramatis pada *maximal margin hyperplane* yang ditampilkan sebagai garis solid. Garis putus-putus menunjukkan *maximal margin hyperplane* yang diperoleh tanpa adanya titik tambahan ini.
- *Maximal margin hyperplane* sangat sensitif terhadap perubahan dalam pengamatan tunggal menunjukkan bahwa mungkin telah overfit data training.

Support Vector Classifier

- Dalam hal ini, kita mungkin bersedia mempertimbangkan classifier berdasarkan hyperplane yang tidak memisahkan dua kelas dengan sempurna, untuk kepentingan
 - ketahanan (robustness) yang lebih besar untuk pengamatan individu,
 - klasifikasi yang lebih baik dari sebagian besar pengamatan data training
- Artinya, mungkin bermanfaat untuk salah mengklasifikasikan beberapa pengamatan data training untuk melakukan pekerjaan yang lebih baik dalam mengklasifikasikan pengamatan yang tersisa (data testing). .
- **Support Vector Classifier**, terkadang disebut *soft margin classifier*, melakukan hal ini dengan tepat.
 - Alih-alih mencari margin sebesar mungkin sehingga setiap pengamatan tidak hanya berada di sisi yang benar dari hyperplane tetapi juga di sisi yang benar dari margin, metode ini membiarkan beberapa pengamatan berada di sisi yang salah dari margin, atau bahkan di sisi yang salah dari hyperplane. (Marginnya lunak (soft) karena bisa dilanggar oleh beberapa pengamatan data training.) ✓

- Pengamatan tidak hanya berada di sisi yang salah dari margin, tetapi juga di sisi yang salah dari hyperplane.
- Padahal, ketika tidak ada hyperplane yang memisahkan, situasi seperti itu tidak bisa dihindari.
- Pengamatan di sisi yang salah dari hyperplane sesuai dengan pengamatan data training yang salah diklasifikasikan oleh support vector classifier.



Tidak ada observasi yang berada di sisi yang salah dari hyperplane.

- *Support vector classifier* mengklasifikasikan pengamatan data testing tergantung pada sisi mana dari hyperplane itu terletak.
- Hyperplane dipilih untuk memisahkan dengan benar sebagian besar observasi data training ke dalam dua kelas, tetapi mungkin salah mengklasifikasikan beberapa observasi.
- Secara singkat, *support vector classifier* adalah solusi untuk masalah optimisasi

$$\underset{\beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n}{\text{maximize}} \quad M$$

subject to $\sum_{j=1}^p \beta_j^2 = 1,$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C,$$

nonnegative tuning parameter

Slack variabel (variabel "kendur") yang memungkinkan pengamatan individu berada di sisi yang salah dari margin atau hyperplane

$\epsilon_i = 0 \rightarrow$ pengamatan ke- i terletak pada sisi margin yang benar ✓

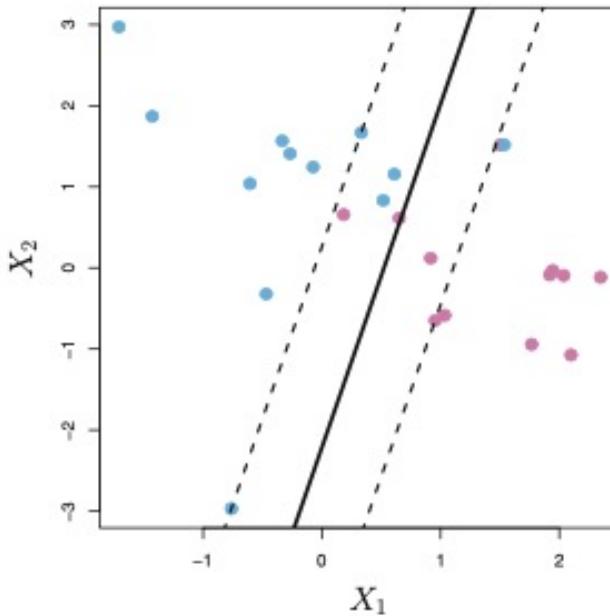
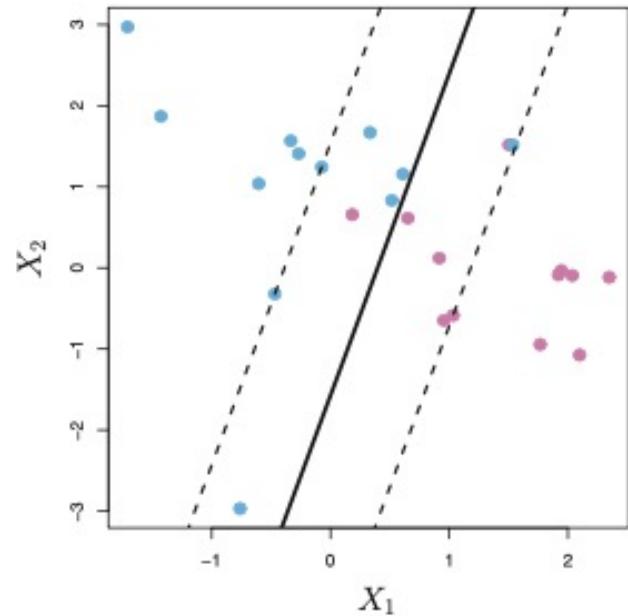
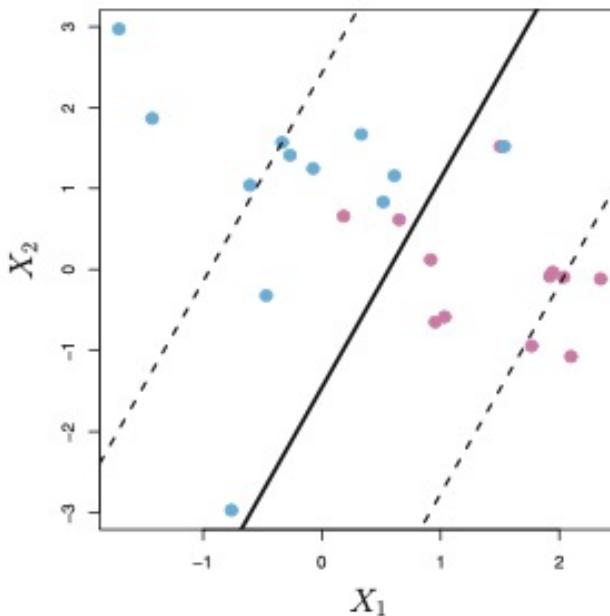
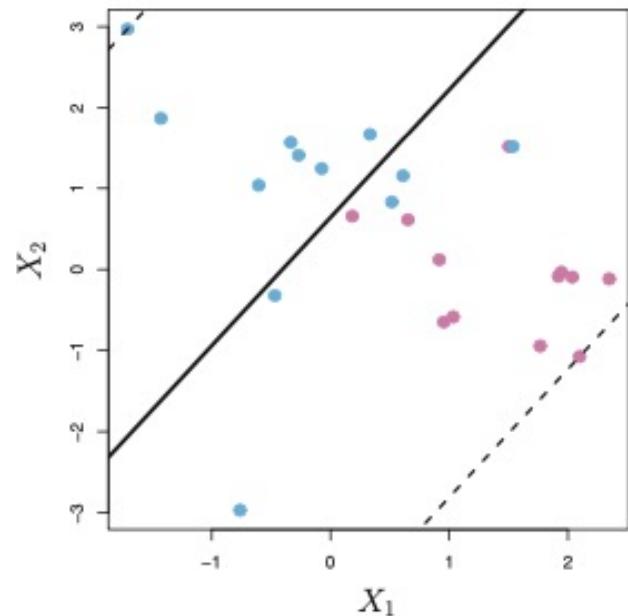
$\epsilon_i > 0 \rightarrow$ pengamatan ke- i terletak pada sisi margin yang salah ✓

$\epsilon_i > 1 \rightarrow$ pengamatan ke- i terletak pada sisi hyperplane yg salah ✓

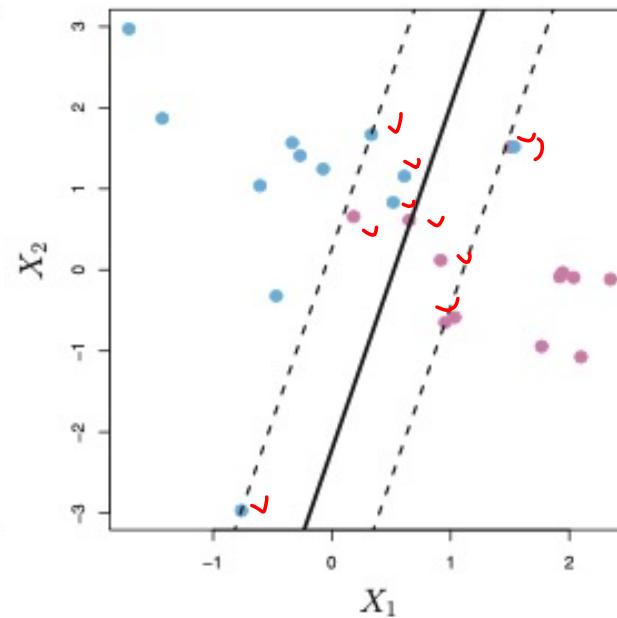
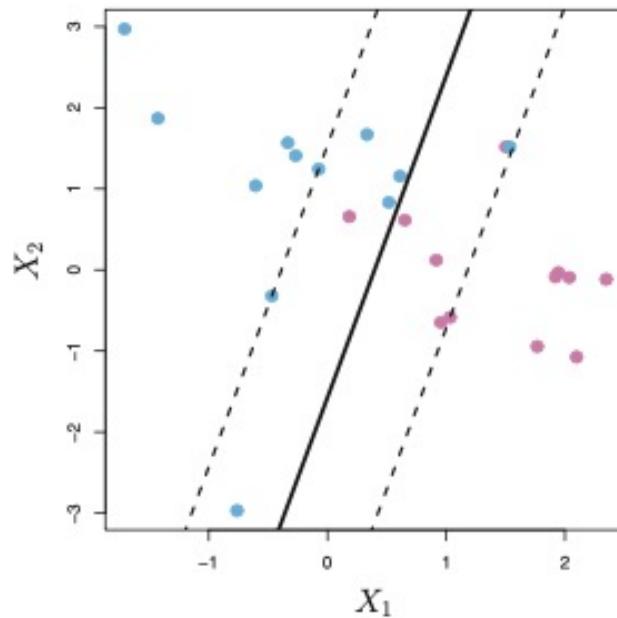
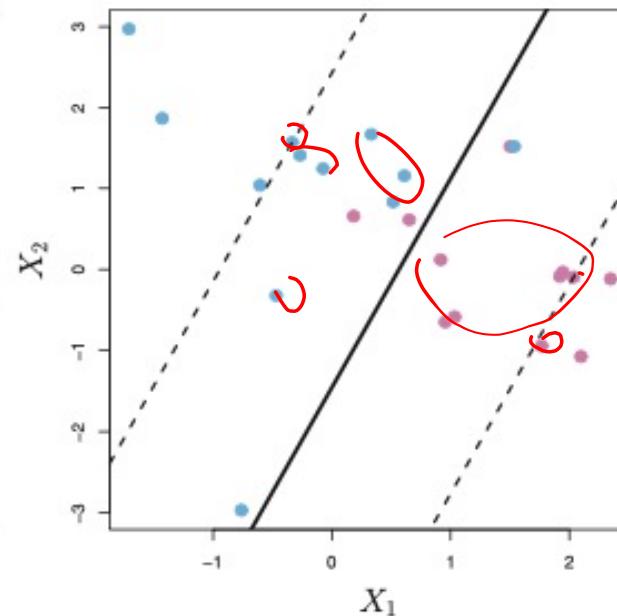
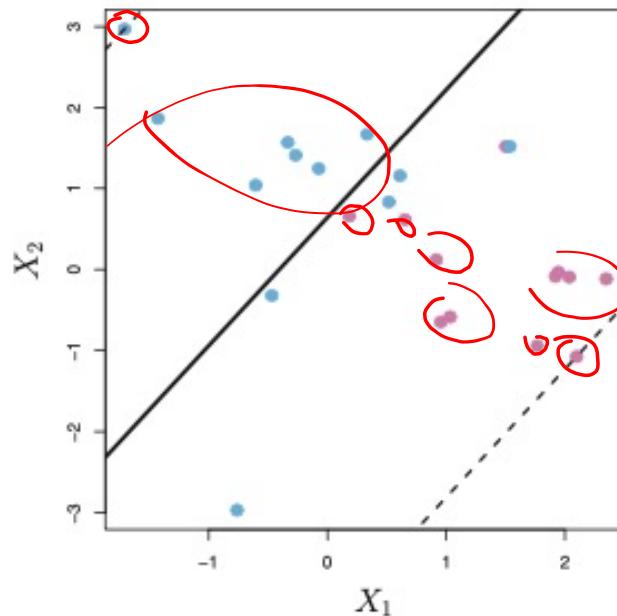
lebar margin; diusahakan membuat kuantitas ini sebesar mungkin

Ketika C kecil, dicari margin sempit yang jarang dilanggar; ini sama dengan pengklasifikasi yang sangat sesuai dengan data, yang mungkin memiliki bias rendah tetapi ragam tinggi. ✓

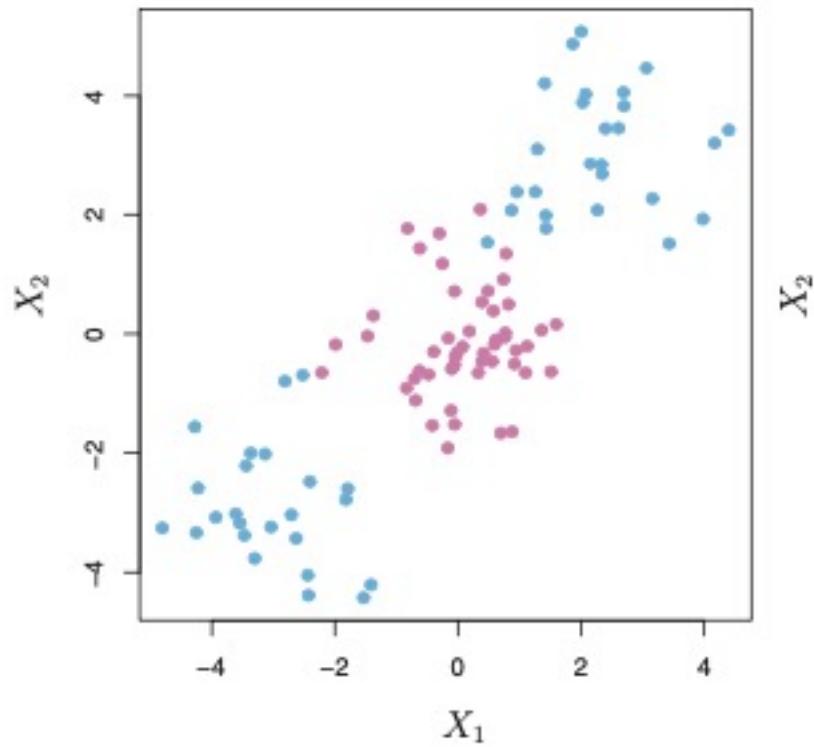
Di sisi lain, ketika C lebih besar, marginnya lebih lebar dan diizinkan lebih banyak pelanggaran; ini sama dengan menyesuaikan data dengan lebih mudah dan mendapatkan pengklasifikasi yang berpotensi lebih bias tetapi mungkin memiliki ragam yang lebih rendah. ✓



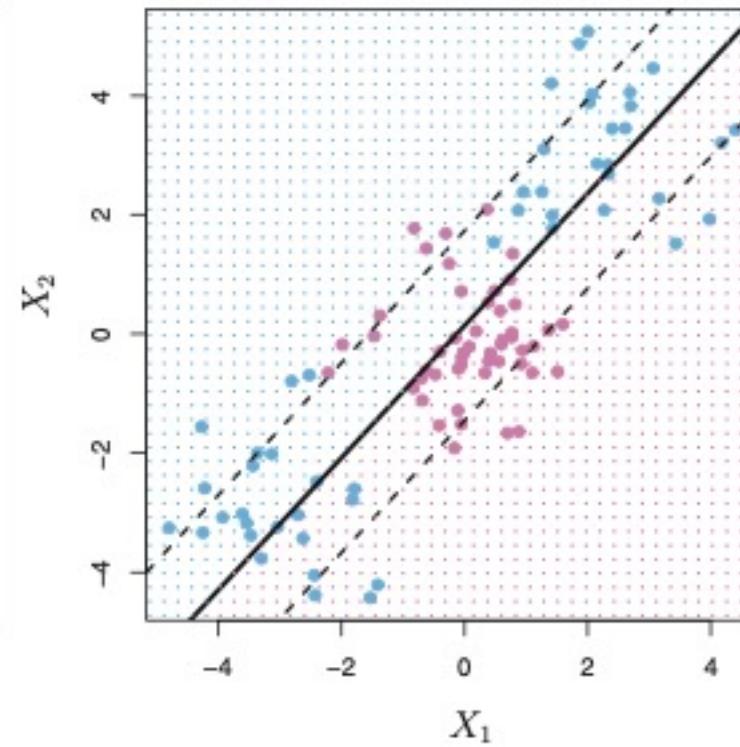
- Pengelompokan *support vector classifier* menggunakan empat nilai berbeda dari tuning parameter C .
- Nilai C terbesar digunakan di panel kiri atas, dan nilai yang lebih kecil digunakan di panel kanan atas, kiri bawah, dan kanan bawah.
- Ketika C besar, maka ada toleransi yang tinggi untuk pengamatan berada di sisi margin yang salah, sehingga marginnya akan besar.
- Saat C berkurang, toleransi untuk pengamatan berada di sisi yang salah dari margin berkurang, dan margin menyempit.



- Pengamatan yang terletak langsung di margin, atau di sisi yang salah dari margin kelasnya, dikenal sebagai vektor pendukung (*support vector*). •
- Panel kiri atas: pengklasifikasi ini memiliki ragam rendah (karena banyak pengamatan adalah vektor pendukung (*support vector*)) tetapi berpotensi memiliki bias yang tinggi. •
- Sebaliknya, jika C kecil, maka akan ada lebih sedikit vektor pendukung (*support vector*) dan karenanya pengklasifikasi yang dihasilkan akan memiliki bias yang rendah tetapi ragam yang tinggi. Panel kanan bawah mengilustrasikan pengaturan ini, dengan hanya delapan vektor pendukung (*support vector*). •



Pengamatan dibagi menjadi dua kelas, dengan batas non-linier di antara keduanya.



Support vector classifier mencari batas linier, dan akibatnya kinerjanya sangat buruk.

Support Vector Machines

- *Support vector classifier* adalah pendekatan alami untuk klasifikasi dalam pengaturan dua kelas, jika batas antara kedua kelas adalah linier. Namun, dalam praktiknya kita terkadang dihadapkan pada batasan kelas yang tidak linier.
 $x_1 \dots x_p \rightarrow$ tinggi nonlinier
- Dalam kasus *support vector classifier*, kita dapat mengatasi masalah kemungkinan batas non-linear antar kelas dengan cara memperbesar ruang fitur menggunakan fungsi polinomial kuadrat, kubik, dan bahkan orde lebih tinggi dari prediktor (fungsi polynomial dari variabel prediktor).
- Tidak sulit untuk melihat bahwa ada banyak cara yang memungkinkan untuk memperbesar ruang fitur, dan jika kita tidak hati-hati, kita bisa mendapatkan banyak fitur. Kemudian perhitungan akan menjadi tidak terkendali. **Support vector machines**, memungkinkan untuk memperbesar ruang fitur yang digunakan oleh *support vector classifier* dengan cara yang mendarah pada komputasi yang efisien.

Support Vector Machines

- **Support vector machine** (SVM) adalah pengembangan dari *support vector classifier* yang dihasilkan dari memperbesar ruang fitur dengan cara tertentu, yakni menggunakan kernel.
- Kita mungkin ingin memperbesar ruang fitur kita untuk mengakomodasi batas non-linear antar kelas. Pendekatan kernel yang diuraikan di sini hanyalah sebuah pendekatan komputasi yang efisien untuk menjalankan ide ini.
- Kernel: fungsi yang mengukur kesamaan dua pengamatan ✓

$$\underline{K(x_i, x_{i'})} = \sum_{j=1}^p x_{ij} x_{i'j} \longrightarrow \text{Kernel linier untuk } \underline{\text{support vector classifier}}$$

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^p x_{ij} x_{i'j} \right)^d \checkmark \longrightarrow \text{Kernel polinomial dengan derajat } \underline{d}$$

- Pada dasarnya sama dengan fitting *support vector classifier* dalam ruang berdimensi lebih tinggi yang melibatkan polinomial derajat d , bukan di ruang fitur aslinya. Ketika *support vector classifier* digabungkan dengan kernel non-linear seperti kernel polinomial, classifier yang dihasilkan dikenal sebagai **support vector machine**.

Support Vector Machines

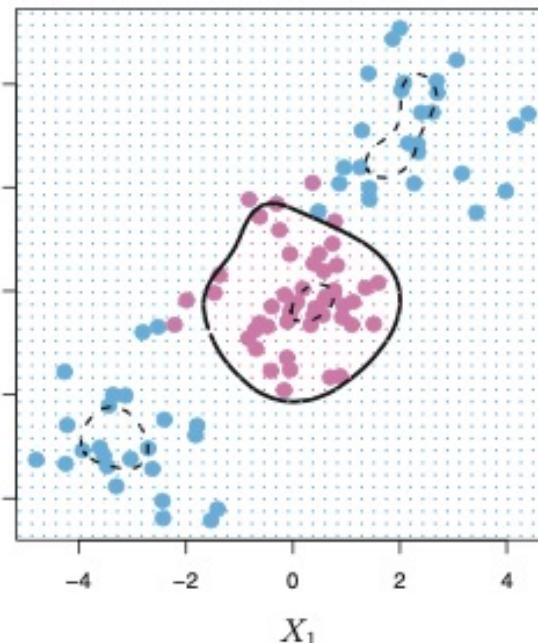
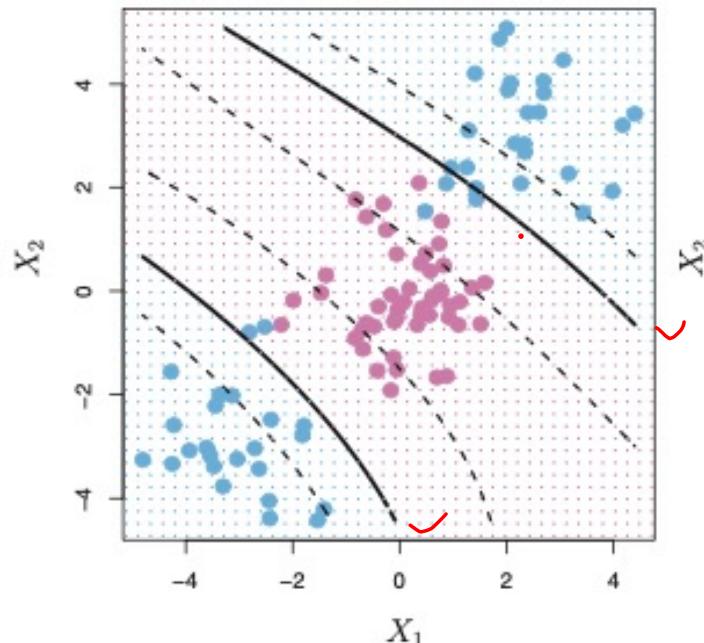
- Sehingga, pada kasus non-linier, fungsi SVM-nya menjadi:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i)$$

- Pilihan kernel lainnya yang popular adalah *radial kernel*

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right)$$

SVM dengan kernel polinomial derajat 3 diterapkan pada data non-linear



$$f(\underline{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

→ support vector classifier

→ SVC dgn (polynomial) ✗

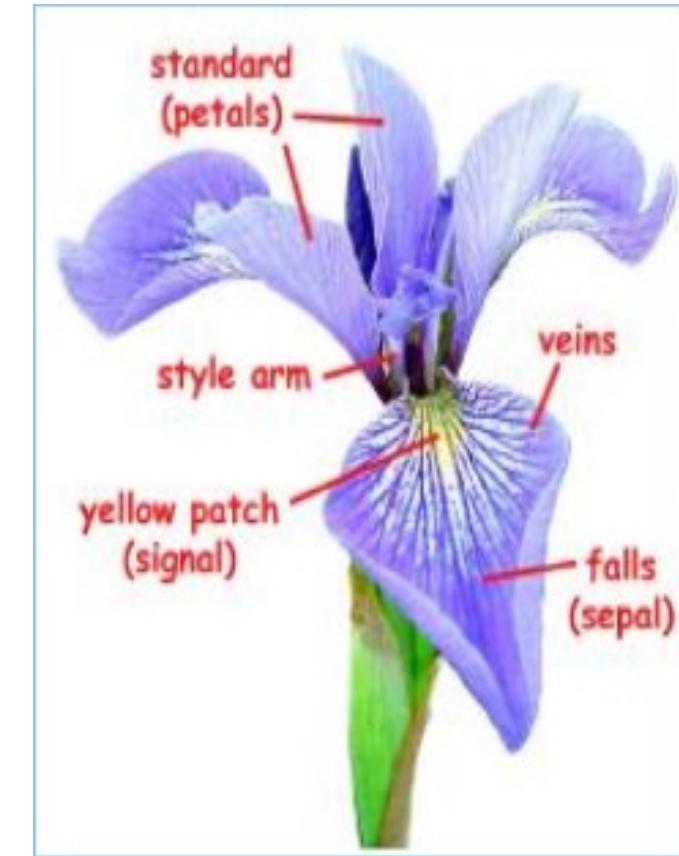
$f(\underline{x}) = (\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p) + \beta_{p+1} x_1^2 + \dots + \beta_{p+k} x_k^2$

→ SVM dgn Kernel

Aplikasi di R

- Using IRIS data set.
- We have features of three kinds of flowers: Setosa, Versicolor, and Virginica.
- We need to discriminate between these three kinds of flower through their features so we will take some of these features as training of SVM and some of these features as testing of this classifier to know the ability of this classifier to discriminate between these three kinds of flowers.

- The features, which are collected of these flowers, are: Sepal length, Sepal width, Petal length, and Petal width.



Load the data set

```
data(iris)
```

```
head(iris)
```

```
# to see the description
```

```
help(iris)
```

.

```
summary(iris)
```

Create training and testing data set

- Make filtration of the iris Data Set by selecting some rows for testing .

```
testidx<-which(1:length(iris[,1])%%5==0)
```

```
testidx
```

```
> testidx<-which(1:length(iris[,1])%%5==0)
> testidx
[1]  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
[16] 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150
```

Create training and testing data set

- Make selection of the rows which will be used to make training of The SVM. If you note that row 10 and 20 is not in the list of training matrix because these lines are selected as part of other rows which will be used for testing .

```
iristrain<-iris[-testidx,]
```

```
iristrain
```

```
head(iristrain)
```

```
> head(iristrain)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1        3.5         1.4        0.2  setosa
2          4.9        3.0         1.4        0.2  setosa
3          4.7        3.2         1.3        0.2  setosa
4          4.6        3.1         1.5        0.2  setosa
6          5.4        3.9         1.7        0.4  setosa
7          4.6        3.4         1.4        0.3  setosa
```

Create training and testing data set

- Make selection of the rows which will be used for testing . If you note that rows 10 and 20 in the list of these rows because these rows are used for testing.

```
iristest<-iris[testidx,]
```

```
head(iristest)
```

```
.
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5	5.0	3.6	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
20	5.1	3.8	1.5	0.3	setosa
25	4.8	3.4	1.9	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa

- Load the package.

```
library(e1071)
```

- Make training of SVM classifier as follows:

```
model <-svm(Species~., data=iristrain)
```

```
print(model)
```

```
summary(model)
```

Make Prediction

- Using the SVM in prediction by testing the model with a new data. In this step, we will see the ability of SVM in discriminating between different types of Data of three kinds of flowers : Setosa , Versicolor and Virginica.

```
prediction<-predict(model,iris[,-5])
```

```
prediction
```

```
> prediction
      5       10       15       20       25
setosa setosa setosa setosa setosa
      30       35       40       45       50
setosa setosa setosa setosa setosa
      55       60       65       70       75
versicolor versicolor versicolor versicolor versicolor
      80       85       90       95      100
versicolor versicolor versicolor versicolor versicolor
     105      110      115      120      125
virginica virginica virginica versicolor virginica
     130      135      140      145      150
virginica virginica virginica virginica virginica
Levels: setosa versicolor virginica
```

Confusion Matrix

- The last Step, we will see whether the SVM classifier can predict with all types of flowers based on the testing data which we gave to it or not.
- As we can see, the SVM classifier can predict with the Setosa and Versicolor flowers perfectly where the classification accuracy is 100% , but it can't predict the Virginica flower perfectly where the classification accuracy is 90%.

```
table(iristest$Species,prediction)
```

```
> table(iristest$Species,prediction)
      prediction
           setosa versicolor virginica
setosa          10         0         0
versicolor        0        10         0
virginica         0         1         9
```

REMINDER Tugas Kelompok – Sesi UTS

- Buatlah proposal penelitian mengenai Projek Kelompok-nya, yang di dalamnya berisi
 1. Judul berdasarkan topik projek yang ditentukan
 2. Latar belakang dan tujuan
 3. Data dan peubah-peubah yang digunakan
 4. Metodologi (rencana tahapan analisis data yang dilakukan).
- Selain proposal penelitian dalam format makalah, dikumpulkan juga file presentasi dalam powerpoint.
- File proposal penelitian dan file presentasi diupload pada form
<https://ipb.link/project-uts-tpm>
- Batas waktu pengiriman adalah **Hari Jumat, tanggal 3 Maret 2023 jam 23:59 WIB**

Terima kasih 😊