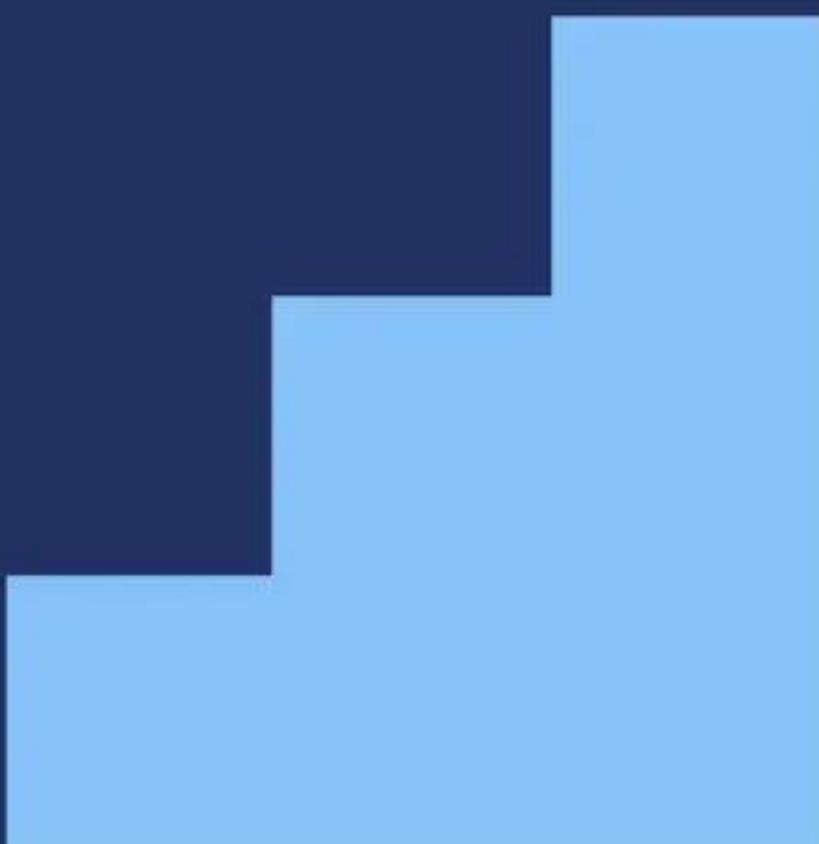


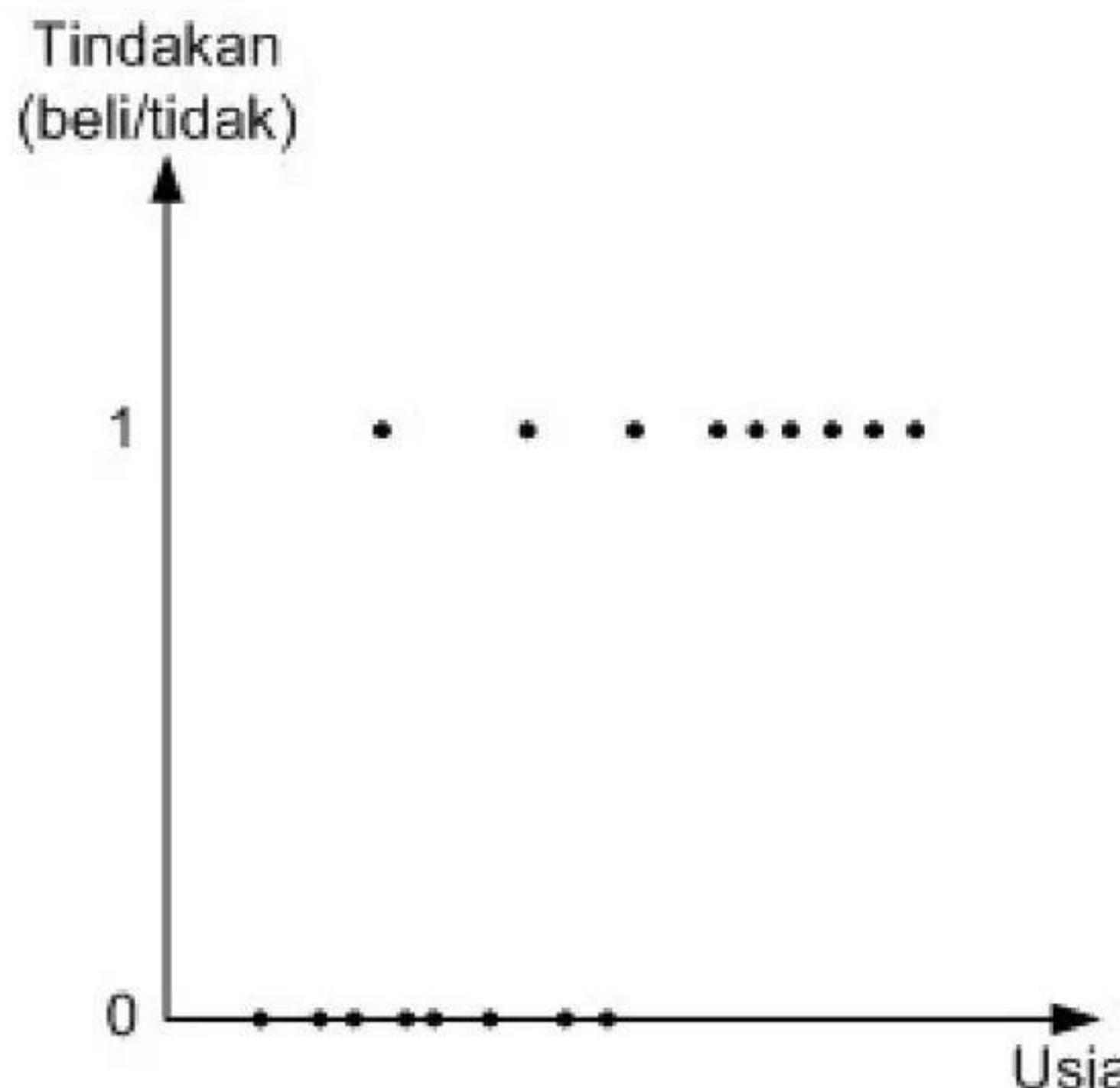
Klasifikasi dengan Regresi Logistik

PERTEMUAN 6



Regresi Logistik

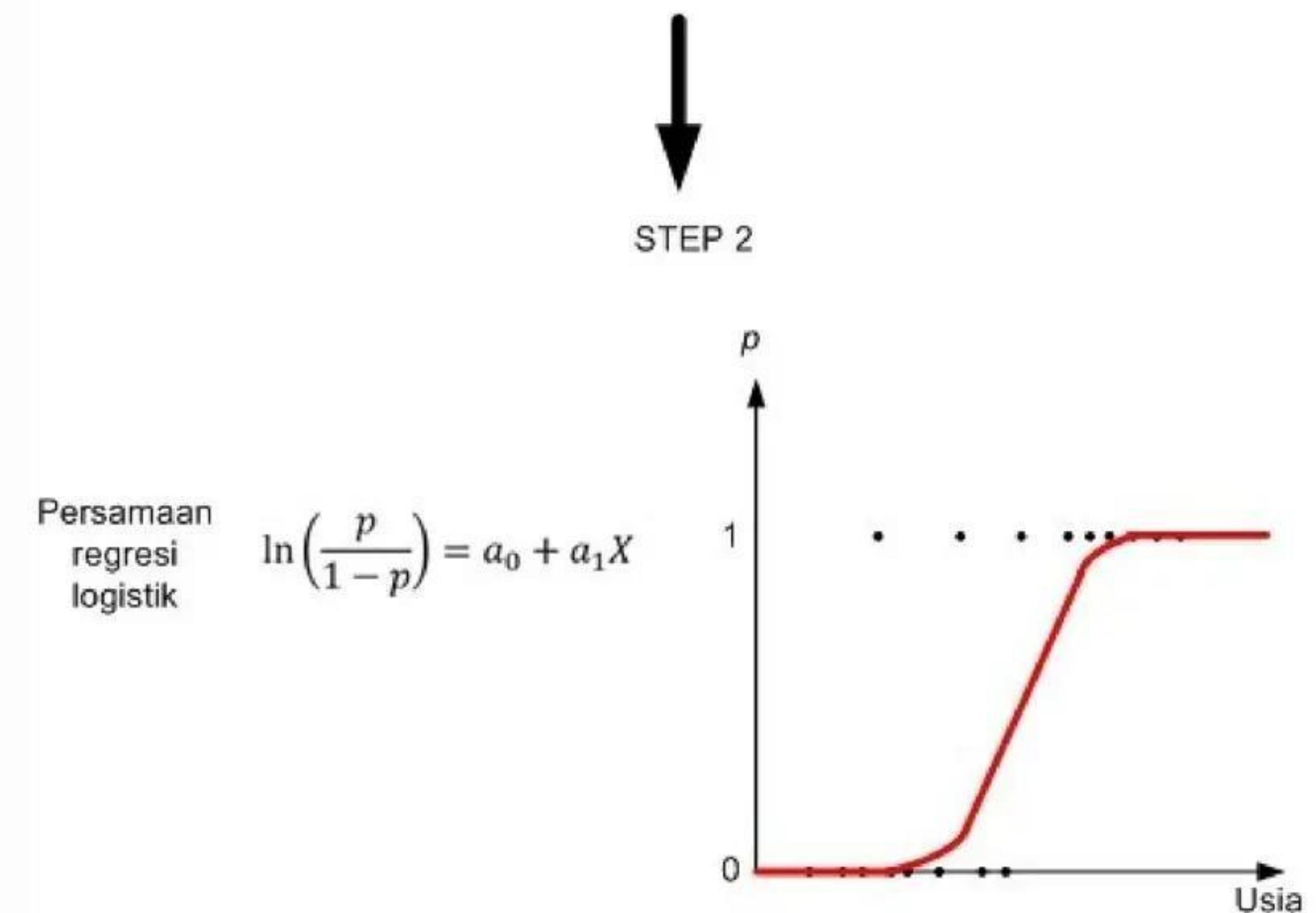
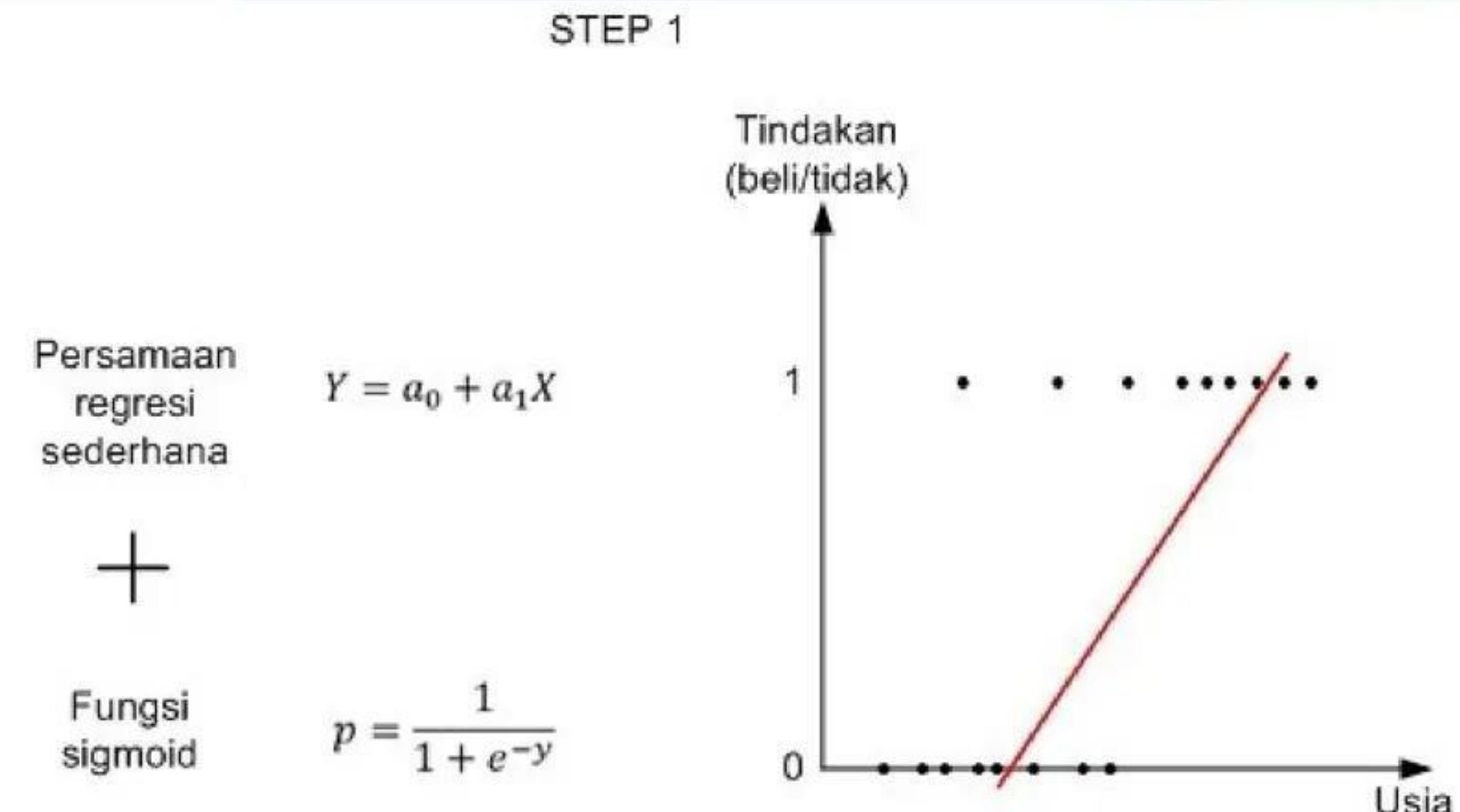
- Berbeda dengan regresi pada umumnya yang memprediksi target numerik, *logistic regression* merupakan teknik regresi yang fungsinya memprediksi target dengan tipe **kategori**.
- Untuk bisa memahami bagaimana regresi logistik bekerja, mari lihat melalui *scatter plot* dari 2 variabel berikut, di mana sumbu x (*feature*) adalah usia, dan sumbu y (*target*) adalah keputusan beli/tidak:



- Terlihat bahwa keputusan seseorang beli/tidak bisa dibedakan berdasarkan usianya.
- *Scatter plot* di samping menunjukkan semakin besar usia, maka kecenderungan untuk beli juga semakin meningkat, namun ada batas di mana usia tertentu kecenderungannya mengecil (seakan-akan ada batas yang bisa memisahkan antara keputusan beli/tidak)

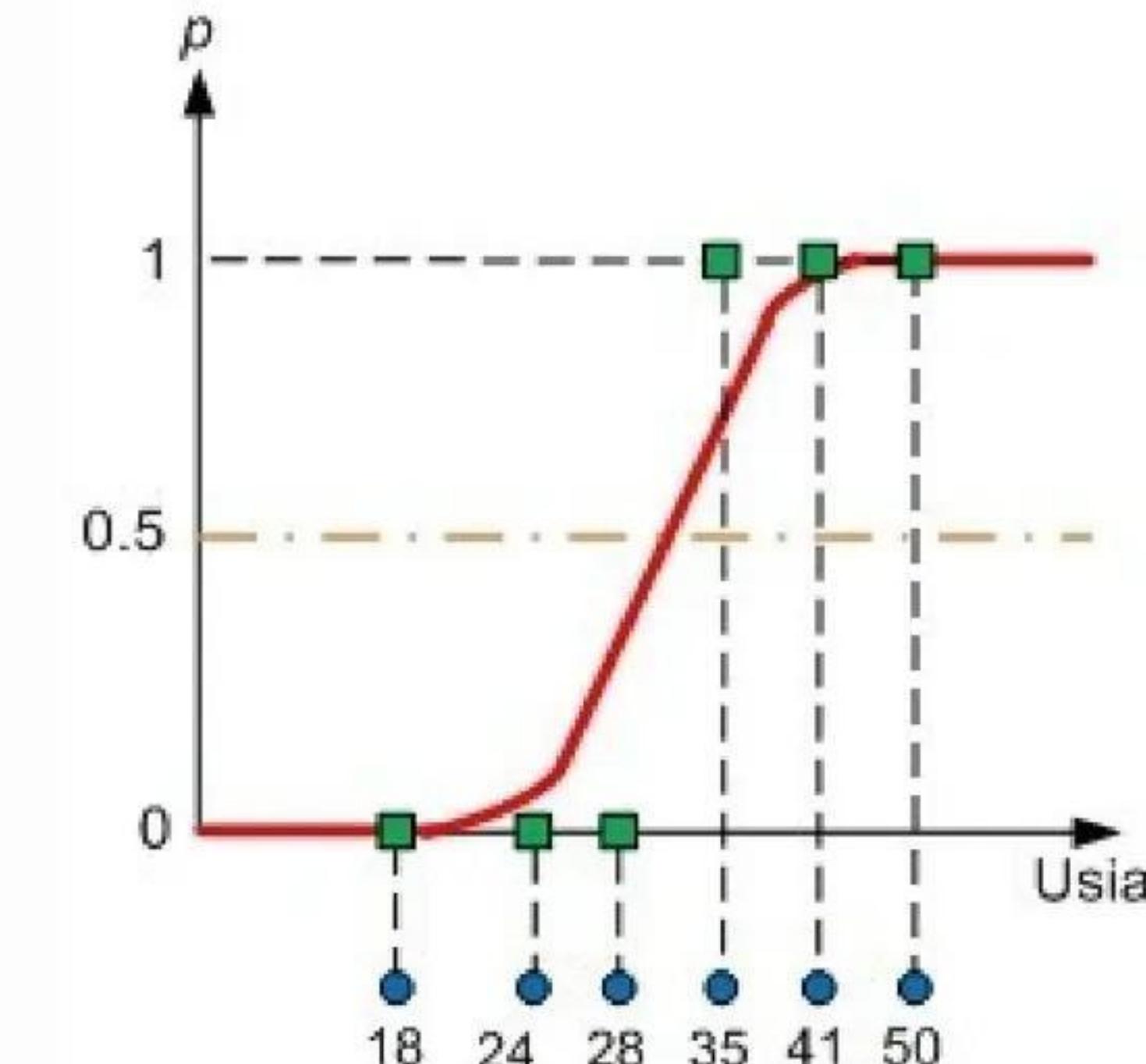
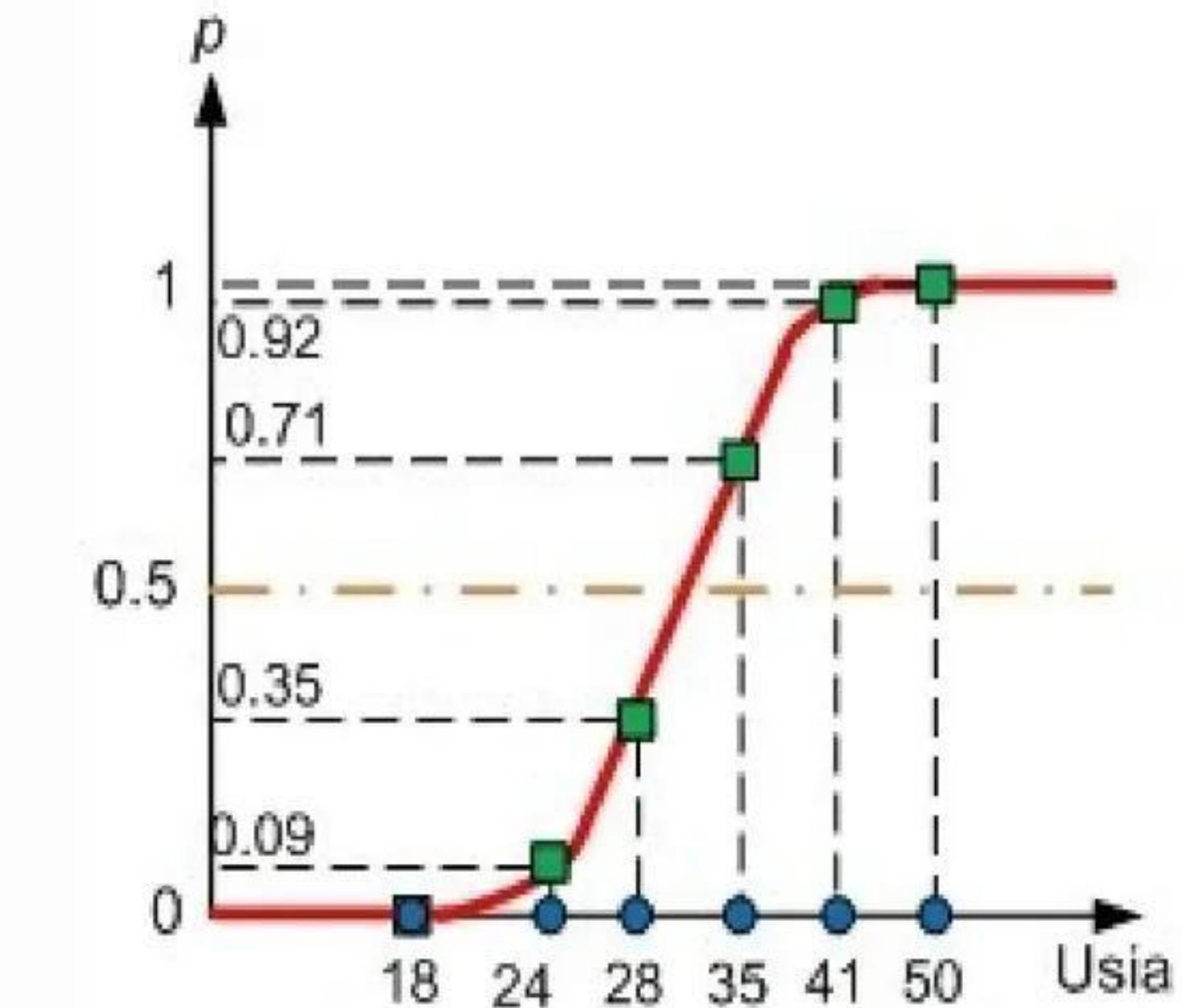
Regresi Logistik (2)

- Jika menggunakan persamaan regresi sederhana untuk kasus ini, maka kurang tepat (ilustrasi di samping).
- Masalah ini sangat cocok diselesaikan dengan *logistic regression*, di mana mengaplikasikan fungsi sigmoid terhadap fungsi awalnya, sehingga kurvanya berbentuk melengkung (ilustrasi di samping).
- Sumbu y yang tadinya adalah keputusan beli/tidak berubah menjadi p yang merupakan probabilitas untuk melakukan keputusan beli.



Regresi Logistik (3)

- Sumbu y (nilai p) adalah probabilitas keputusan beli.
- Setelah itu kita tentukan batas (threshold), di mana nilai batas ini biasanya adalah 0,5 (tepat di tengah-tengah).
- Semua nilai x (usia) yang diproyeksikan ke p , akan menjadi bernilai 1 jika ia memiliki nilai $\geq 0,5$ dan bernilai 0 jika ia memiliki nilai $< 0,5$ (ilustrasi di samping).
- Dapat dikatakan bahwa usia 28 (nilai $p=0,35$) akan dikategorikan sebagai keputusan tidak beli (0), sementara usia 41 (nilai $p=0,92$) akan dikategorikan sebagai keputusan beli (1).
- Tentunya bisa mengubah-ubah nilai batas (0,5) sesuai kehendak yang nantinya akan mempengaruhi hasil prediksi model.



Mengapa Regresi Logistik Disebut Regresi Logistik dan Bukan Klasifikasi Logistik?

Mengapa kita tidak menyebutnya "Klasifikasi Logistik"?

- Regresi logistik menggunakan rumus dasar yang sama dengan regresi linier tetapi regresi ini dilakukan untuk mengetahui kemungkinan hasil kategorikal.
- Regresi linier memberikan nilai kontinu keluaran y untuk masukan tertentu X. Sedangkan regresi logistik memberikan nilai kontinu $P(Y=1)$ untuk masukan tertentu X, yang kemudian diubah menjadi Y=0 atau Y=1 berdasarkan sebuah nilai ambang batas.

Rumusnya:

Dalam regresi linier, memperkirakan variabel keluaran Y berdasarkan jumlah tertimbang dari variabel masukan.

Rumusnya adalah sebagai berikut:

$$Y = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n$$

Where,

Y = Dependent Variable (DV)

X_1, X_2, X_n = Independent Variable (IV)

b_0 = Y intercept

b_1, b_2, b_n = Coefficient of slope

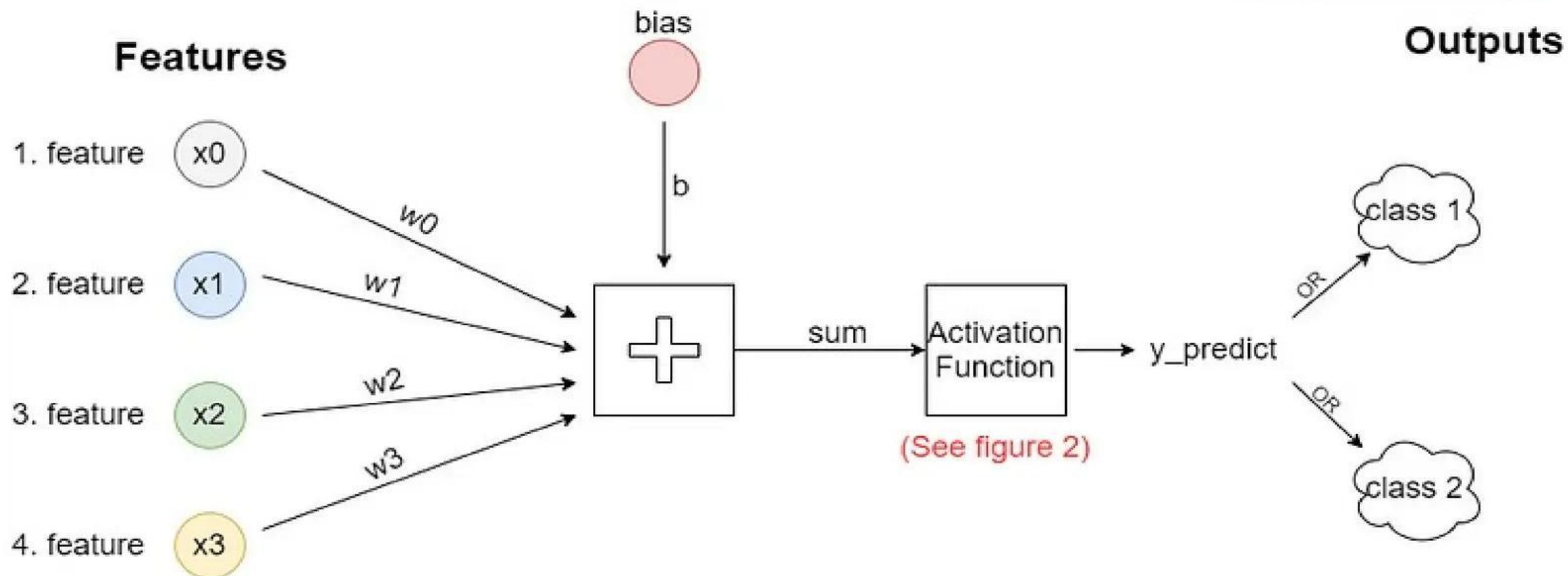
Dalam regresi logistik, dilakukan hal yang persis sama dengan regresi linier tetapi dengan sedikit tambahan. Dimana meneruskan hasilnya melalui fungsi khusus yang dikenal sebagai **Fungsi Sigmoid** untuk memprediksi keluaran Y.

$$Y = \text{Sigmoid}(b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n)$$

$$\text{where, Sigmoid} = f(x) = \frac{1}{1 + e^{-x}}$$

$$\text{Therefore, } Y = \frac{1}{1 + e^{-(b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n)}}$$

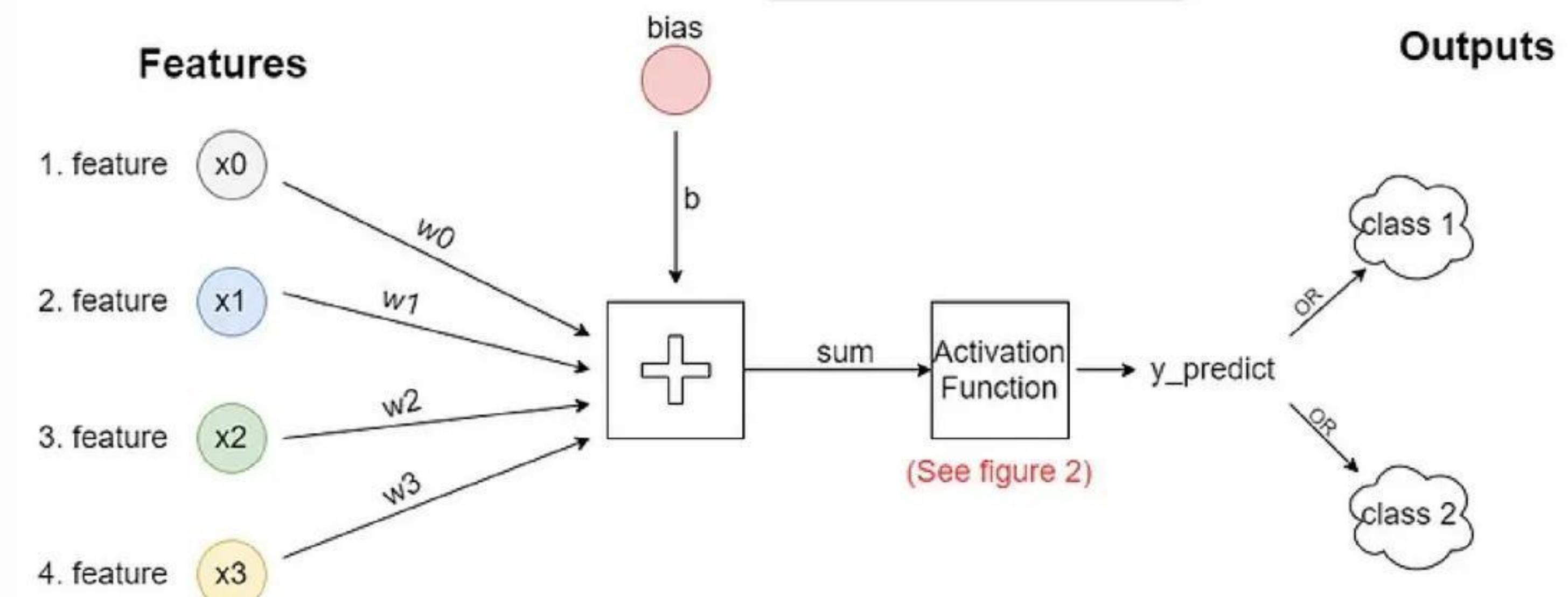
Algoritma Regresi Logistik



- ▶ Weight: coefficient of each feature
- ▶ Bias: intercept
- ▶ $\text{sum} = x_0 \cdot w_0 + x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 + b$
- ▶ $y_{\text{predict}} = \text{Activation Function}(\text{sum})$

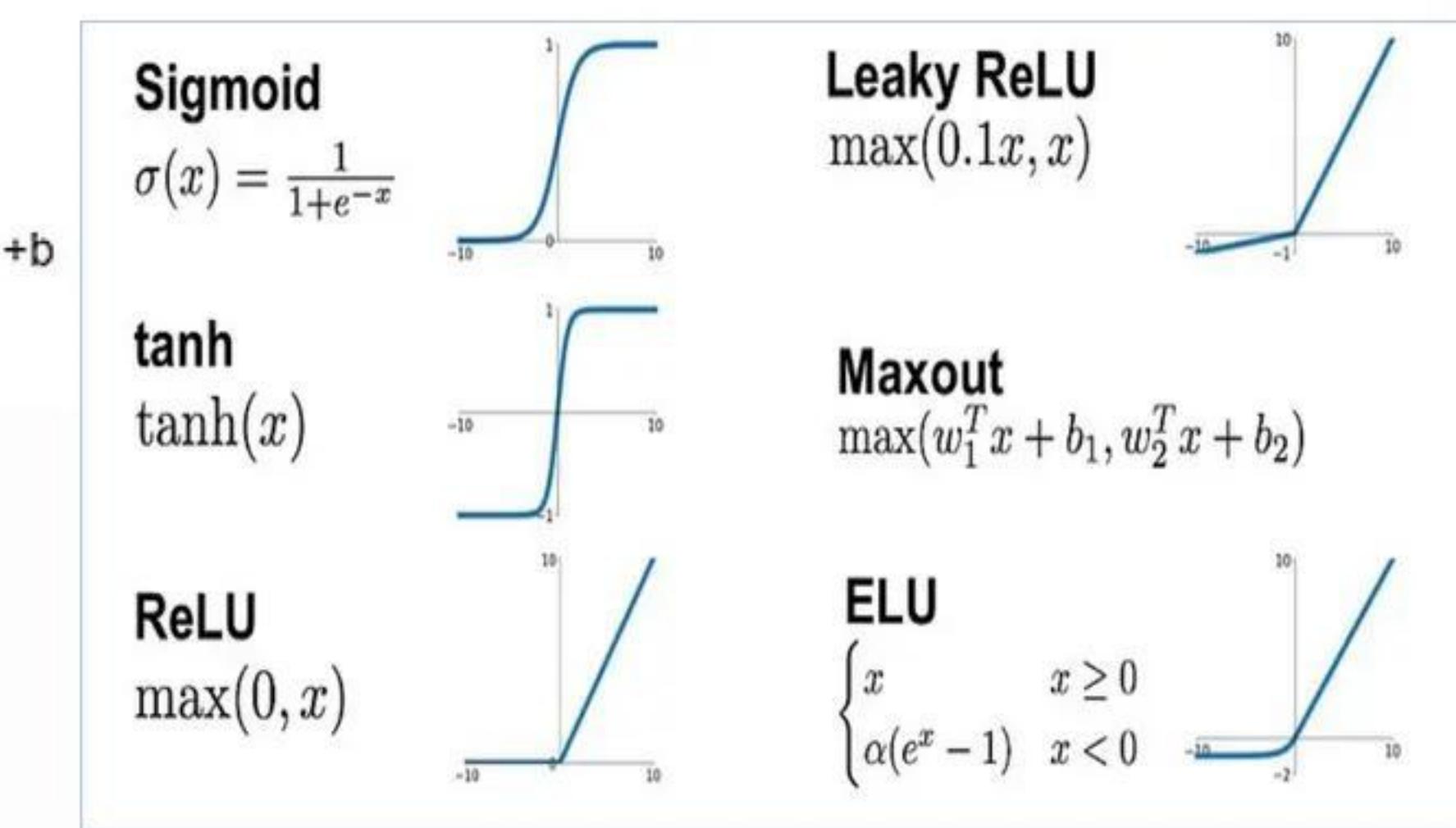
Algoritma Regresi Logistik (2)

1. Pada bagian pelatihan (Training) regresi logistik, 1 bobot diberikan untuk setiap fitur dan 1 nilai bias ditambahkan ke sistem.
2. Nilai bobot dan bias ini diinisialisasi dengan nilai yang dipilih oleh pengguna.
3. Setiap bobot dikalikan dengan nilai fiturnya sendiri dan dijumlahkan dengan menjumlahkan nilai biasnya.
4. Fungsi aktivasi (gambar 2) yang dipilih oleh pengguna diterapkan pada nilai penjumlahan.
5. Jika kita berasumsi bahwa ini adalah fungsi sigmoid, Fungsi Aktivasi (jumlah) akan menjadi antara 0-1 dan kesalahan dihitung dengan nilai ini.
6. Berdasarkan nilai kesalahan yang dihitung, nilai bobot dan bias diperbarui dengan penurunan gradien sesuai dengan kecepatan pembelajaran.
7. Proses ini diulangi sebanyak jumlah iterasi.



- Weight: coefficient of each feature
- Bias: intercept
- $\text{sum} = x_0 * w_0 + x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + b$
- $y_{\text{predict}} = \text{Activation Function}(\text{sum})$

Gambar 1. Diagram Alir Gambar Regresi Logistik



Gambar 2. Fungsi Aktivasi

Contoh Regresi Logistik

Beberapa kasus di mana algoritma klasifikasi Regresi Logistik digunakan adalah sebagai berikut :

1. Untuk mengetahui apakah email yang diterima merupakan email spam atau bukan.
2. Untuk mengetahui apakah pinjaman bank dapat diberikan kepada nasabah tertentu atau tidak.
3. Untuk mengetahui apakah seorang anak akan lulus atau gagal dalam suatu ujian.
4. Untuk mengetahui apakah seseorang menderita diabetes atau tidak.
5. Untuk mengidentifikasi apakah sentimen suatu kalimat positif atau negatif.

Metrik Klasifikasi - confusion matrix

CONFUSION MATRIX

		Predicted Category		
		0	1	Total
Actual category	0	Truenegatives: Predicted 0 Actually 0	Falsepositives: Predicted 1 Actually 0	Total actually negative
	1	Falsenegatives: Predicted 0 Actually 1	Truepositives: Predicted 1 Actually 1	Total actually positive
Total		Total Predictednegative	Total Predictedpositive	Grandtotal

9

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FN} + \text{FP} + \text{TP}} = \frac{\text{TN} + \text{TP}}{N}$$

$$\text{Proportion of true positives} = \text{PTP} = \frac{\text{TP}}{\text{TPP}} = \frac{\text{TP}}{\text{FP} + \text{TP}}$$

$$\text{Sensitivity} = \frac{\text{Number of true positives}}{\text{Total actually positive}} = \frac{\text{TP}}{\text{TAP}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Proportion of true negatives} = \text{PTN} = \frac{\text{TN}}{\text{TPN}} = \frac{\text{TN}}{\text{FN} + \text{TN}}$$

$$\text{Specificity} = \frac{\text{Number of true negatives}}{\text{Total actually negative}} = \frac{\text{TN}}{\text{TAN}} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

$$F1 = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Metrik Klasifikasi – confusion matrix



Metrik Klasifikasi – Kurva ROC

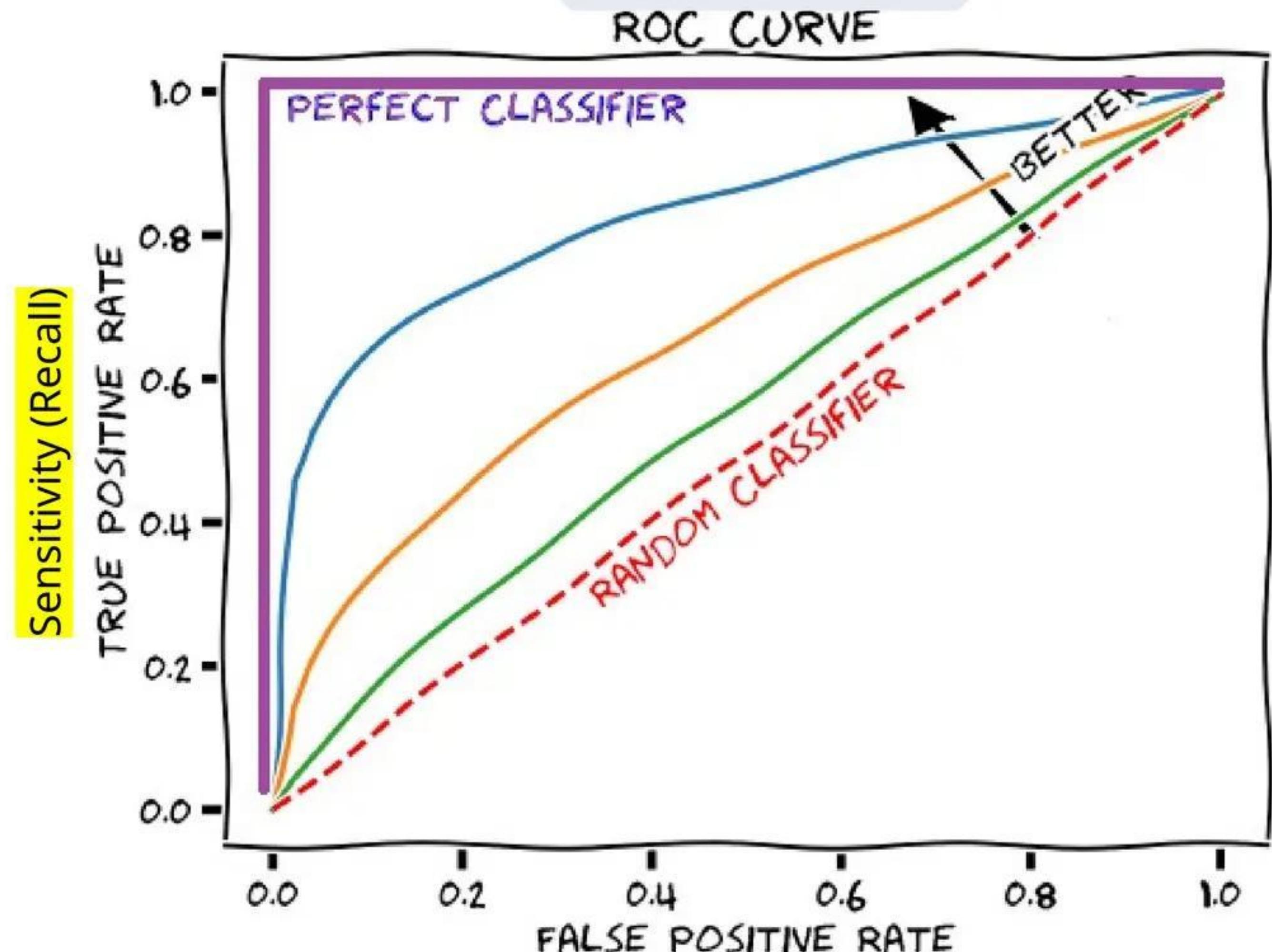
$$\text{Sensitivity} = \frac{\text{Number of true positives}}{\text{Total actually positive}} = \frac{TP}{TAP} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{\text{Number of true negatives}}{\text{Total actually negative}} = \frac{TN}{TAN} = \frac{TN}{FP + TN}$$

$$\text{False positive rate} = 1 - \text{specificity} = \frac{FP}{TAN} = \frac{FP}{FP + TN}$$

$$\text{False negative rate} = 1 - \text{sensitivity} = \frac{FN}{TAP} = \frac{FN}{TP + FN}$$

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive



- **ROC** (Receiver Operator Characteristic) menggambarkan semua kemungkinan performa classifier untuk berbagai jenis batas (threshold), sementara nilai accuracy hanya untuk 1 threshold.
- **AUC** (Area under the Curve) adalah luas permukaan di bawah kurva ROC.

Accuracy

Jumlah prediksi yang dibuat oleh model dengan benar (baik yang bernilai positif maupun negatif). Semakin tinggi nilai akurasi yang dihasilkan oleh model, maka ketepatan nilai prediksi akan semakin mendekati nilai aktual (nilai sebenarnya).

- pred MACET- true MACET: Jumlah data yang diprediksi macet dan kenyataannya macet (**TP**)
- pred LANCAR-true LANCAR: Jumlah data yang diprediksi lancar dan kenyataannya lancar (**TN**)
- pred MACET-true LANCAR: Jumlah data yang diprediksi macet tapi kenyataannya lancar (**FP**)
- pred LANCAR-true MACET: Jumlah data yang diprediksi lancar tapi kenyataannya macet (**FN**)

		accuracy: 90.00%			class precision
		true MACET	true LANCAR		
pred. MACET	53	4	92.98%		
	6	37	86.05%		
class recall		89.83%	90.24%		

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{53 + 37}{53 + 37 + 4 + 6} = \frac{90}{100} = 90\%$$

Berdasarkan perhitungan, akurasi yang dihasilkan oleh model sebesar 90%. Jadi, jika model yang dimiliki saat ini diterapkan pada 100 data baru, hasilnya, model akan memprediksi dengan benar (ideal atau tidak idealnya) sebanyak 90 orang dan 10 orang lainnya bernilai error.

Precision and Recall, and F-measures

- **Precision:** ukuran kebenaran yang dicapai dalam prediksi yang benar (positif).
- **Recall (Sensitivity):** ukuran pengamatan actual yang diprediksi dengan benar.
 - Perfect score is 1.0
 - Inverse relationship between precision & recall
- **Specificity:** ukuran kebenaran yang dicapai dalam prediksi yang benar (negatif).
- **F measure (F1 or F-score):** matriks untuk mengevaluasi model klasifikasi dengan menggunakan nilai ratio rata-rata recall dan precision.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{\text{Number of 'True Negatives'}}{\text{Number of 'True Negatives' + Number of 'False Positives'}}$$

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Tips memilih Performance Metrics

- Jika data set yang digunakan memiliki jumlah True Positive dan True Negative yang seimbang, pilihlah algoritma yang memiliki akurasi tertinggi. Namun jika tidak seimbang maka gunakan F1 score untuk mengevaluasi model.
- Gunakan algoritma yang memiliki precision tinggi jika ditemukan False Positive yang banyak pada dataset.
- Pilih algoritma yang memiliki recall tinggi jika ditemukan Flase Negative yang banyak pada dataset.
- Pilih algoritma yang memiliki specitify tinggi yang memiliki True Negative sebagai prioritas.

Regresi Logistik dengan Scikit-Learn

Dataset : Heart Attack Analysis & Prediction Dataset

Sumber : <https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/data>

Tujuan : ingin memprediksi apakah seseorang kemungkinan terkena serangan jantung.

Dataset terdiri dari 303 record dengan 13 atribut independen dan 1 target.

Atribut:

- Usia : Usia pasien
- Jenis Kelamin : Jenis kelamin pasien (1=laki2, 0=perempuan)
- exang: angina akibat olahraga (1 = ya; 0 = tidak)
- caa: jumlah pembuluh arteri (0-3)
- cp : Tipe nyeri dada
 - Nilai 1 : angina tipikal
 - Nilai 2: angina atipikal
 - Nilai 3: nyeri non-angina
 - Nilai 4: tanpa gejala
- trtbps : tekanan darah istirahat (dalam mm Hg)

- chol : kolesterol dalam mg/dl diambil melalui sensor BMI
- fbs : (gula darah puasa >120 mg/dl)
(1 = benar; 0 = salah)
- rest_ecg : hasil elektrokardiografi istirahat
 - Nilai 0: biasa
 - Nilai 1 : mengalami kelainan gelombang ST-T (inversi gelombang T dan/atau elevasi atau depresi ST > 0,05 mV)
 - Nilai 2: menunjukkan kemungkinan atau pasti hipertrofi ventrikel kiri menurut kriteria Estes
- thalach : tercapai denyut jantung maksimal

Output (Target)

- 0= lebih kecil kemungkinan terkena serangan jantung
- 1= lebih besar kemungkinan terkena serangan jantung

Regresi Logistik dengan Scikit-Learn

Importing Packages and Working with the Data

```
In [35]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix
```

```
In [2]: df = pd.read_csv('heart.csv')
df
```

Out[2]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	sip	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Regresi Logistik dengan Scikit-Learn

Importing and Working with the Data

```
In [3]: df.describe()
```

Out[3]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	sip	caa	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3



Regresi Logistik dengan Scikit-Learn

Importing and Working with the Data

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trtbps    303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalachh  303 non-null    int64  
 8   exng     303 non-null    int64  
 9   oldpeak   303 non-null    float64 
 10  slp       303 non-null    int64  
 11  caa       303 non-null    int64  
 12  thall     303 non-null    int64  
 13  output    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: age          0
         sex          0
         cp           0
         trtbps       0
         chol          0
         fbs          0
         restecg      0
         thalachh     0
         exng          0
         oldpeak       0
         slp           0
         caa           0
         thall          0
         output         0
         dtype: int64
```



Tidak ada missing value

Regresi Logistik dengan Scikit-Learn

Splitting the dataset

Definisikan terlebih dahulu variabel bebasnya yaitu variabel 'X', dan variabel terikatnya yaitu variabel 'Y'.

```
In [5]: # Separating Independent and dependent variables  
X=df.drop(['output'],axis=1)  
Y=df['output']  
X.shape, Y.shape
```

```
Out[5]: ((303, 13), (303,))
```

Dengan menggunakan fungsi 'StandardScaler' di scikit-learn\, kita akan menormalkan variabel independen atau variabel 'X'.

```
In [9]: scaler = StandardScaler()  
X = pd.DataFrame(scaler.fit_transform(X), columns = X.columns.values)  
X
```

```
Out[9]:
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	sip	caa	thall
0	0.952197	0.681005	1.973123	0.763956	-0.256334	2.394438	-1.005832	0.015443	-0.696631	1.087338	-2.274579	-0.714429	-2.148873
1	-1.915313	0.681005	1.002577	-0.092738	0.072199	-0.417635	0.898962	1.633471	-0.696631	2.122573	-2.274579	-0.714429	-0.512922
2	-1.474158	-1.468418	0.032031	-0.092738	-0.816773	-0.417635	-1.005832	0.977514	-0.696631	0.310912	0.976352	-0.714429	-0.512922
3	0.180175	0.681005	0.032031	-0.663867	-0.198357	-0.417635	0.898962	1.239897	-0.696631	-0.206705	0.976352	-0.714429	-0.512922
4	0.290464	-1.468418	-0.938515	-0.663867	2.082050	-0.417635	0.898962	0.583939	1.435481	-0.379244	0.976352	-0.714429	-0.512922
...
298	0.290464	-1.468418	-0.938515	0.478391	-0.101730	-0.417635	0.898962	-1.165281	1.435481	-0.724323	-0.649113	-0.714429	1.123029
299	-1.033002	0.681005	1.973123	-1.234996	0.342756	-0.417635	0.898962	-0.771706	-0.696631	0.138373	-0.649113	-0.714429	1.123029
300	1.503641	0.681005	-0.938515	0.706843	-1.029353	2.394438	0.898962	-0.378132	-0.696631	2.036303	-0.649113	1.244593	1.123029
301	0.290464	0.681005	-0.938515	-0.092738	-2.227533	-0.417635	0.898962	-1.515125	1.435481	0.138373	-0.649113	0.265082	1.123029
302	0.290464	-1.468418	0.032031	-0.092738	-0.198357	-0.417635	-1.005832	1.064975	-0.696631	-0.896862	-0.649113	0.265082	-0.512922

303 rows × 13 columns

Regresi Logistik dengan Scikit-Learn

Splitting the dataset

Selanjutnya, kita membagi data dalam train dan test set. Train set digunakan untuk melatih model regresi logistik. Model belajar dari fitur-fitur yang disertakan dalam training set. Test set digunakan untuk memvalidasi kinerja model regresi logistik. Kita membagi data sehingga train set terdiri dari 80% data, dan test set terdiri dari 20% data.

```
In [10]: # Split the dataset into training and testing sets  
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

Pembuatan Model Regresi Logistik

Dengan memanfaatkan LogisticRegression modul dalam paket scikit-learn, kita dapat menyesuaikan model regresi logistik, menggunakan fitur yang disertakan dalam X_train, ke train set. Untuk melatih model, fungsi fit() digunakan dengan memasukan train set yang sudah disiapkan.

```
In [11]: model = LogisticRegression()  
model.fit(X_train, y_train)
```

```
Out[11]:  
+ LogisticRegression  
| LogisticRegression()
```

Regresi Logistik dengan Scikit-Learn

Pembuatan Model Regresi Logistik

Pada bagian ini, dapat menampilkan koefisien untuk masing-masing dari fitur yang terdapat pada train set.

```
In [20]: print(model.coef_)
```

```
[[ -0.07653786 -0.85844456  0.78464861 -0.1973674 -0.24089346 -0.13433343  
  0.09092294  0.49998533 -0.4756523 -0.63029188  0.13306084 -0.8782799  
 -0.45213954]]
```

Selanjutnya, setelah kita melatih model regresi logistik pada train set, kita dapat menggunakan model tersebut untuk memprediksi apakah orang-orang yang termasuk dalam test set lebih kecil atau besar terkena serangan jantung :

```
In [19]: y_pred = pd.Series(model.predict(X_test))  
y_test = y_test.reset_index(drop=True)  
z = pd.concat([y_test, y_pred], axis=1)  
z.columns = ['True', 'Prediction']  
z.head()
```

Out[19]:

	True	Prediction
0	0	0
1	1	1
2	0	1
3	0	0
4	1	0

Di sini, kita dapat melihat bahwa untuk lima observasi pertama pada test set, model regresi logistik memperkirakan 3 dari 5 dengan benar.

Di baris pertama, ada pasien usia 63 tahun, jenis kelamin laki2, nyeri dada non-angina, dll. diprediksi lebih kecil kemungkinan terkena serangan jantung oleh model. Ternyata benar bahwa pasien dengan gejala2 tersebut lebih kecil kemungkinan terkena serangan jantung.

Regresi Logistik dengan Scikit-Learn

Mengukur kinerja model

Untuk mengevaluasi seluruh rangkaian pengujian, kita dapat menggunakan metricsmodul dari paket scikit-learn.

```
In [36]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print("Precision:", metrics.precision_score(y_test, y_pred))
print("Recall:", metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8524590163934426
Precision: 0.8378378378378378
Recall: 0.9117647058823529
```

Artinya, model regresi logistik menghasilkan akurasi sebesar 85,2%. Model dapat dikatakan berkinerja cukup baik. Angka precision 83,8% menunjukkan bahwa semua kemunculan data yang positif berhasil "ditangkap".

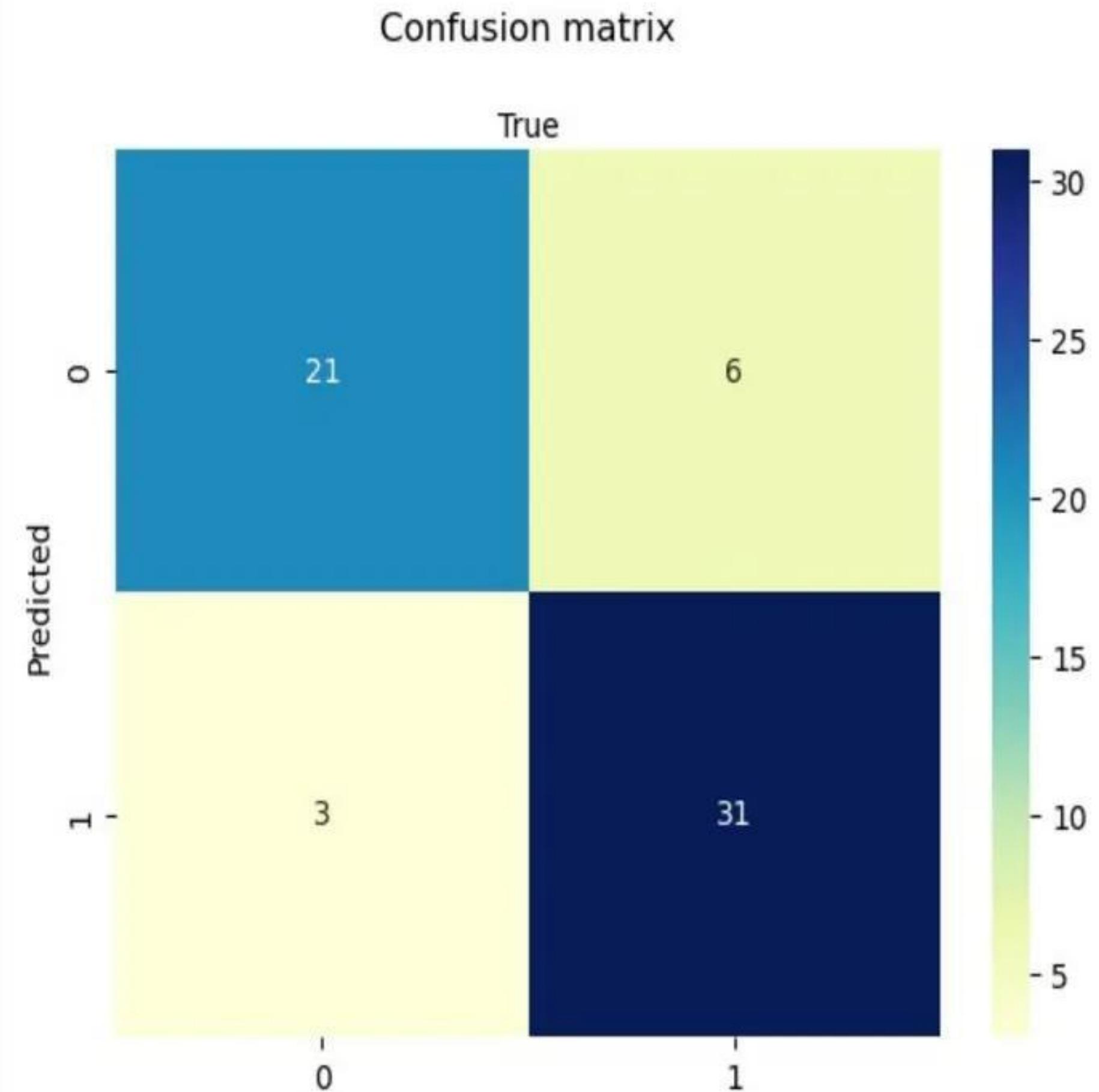
Regresi Logistik dengan Scikit-Learn

Mengukur kinerja model

```
In [56]: cnf_matrix = metrics.confusion_matrix(y_test, y_pred)

labels = [0, 1]
fig, ax = plt.subplots()
tick_marks = np.arange(len(labels))
plt.xticks(tick_marks, labels)
plt.yticks(tick_marks, labels)
# create heatmap
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu", fmt='g')
ax.xaxis.set_label_position("top")
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Predicted')
plt.xlabel('True')
```

Angka TP = 21 dan FP = 6 artinya model memprediksi 21 kasus positif yang tepat, dan 6 kasus positif yang tidak tepat. Demikian pula FN = 3 dan TN = 31 berarti model memprediksi 3 kasus negatif yang tepat dan 31 kasus negatif yang keliru.



Latihan

Seorang peneliti ingin mengetahui bagaimana pengaruh kualitas pelayanan publik terhadap kepuasan pengguna (masyarakat). Kualitas pelayanan publik diteliti melalui uji variabel Daya Tanggap (X1) dan Empati (X2). Kepuasan penggunaan layanan (Y) sebagai variabel dependen adalah variabel dummy dimana jika responden menjawab puas maka kita beri skor 1 dan jika menjawab tidak puas kita beri skor 0.

Analisislah data berikut dengan regresi logistik !

No	X1	X2	Y	No	X1	X2	Y	No	X1	X2	Y
1	31	46	1	18	34	43	1	35	34	40	1
2	33	38	1	19	33	40	0	36	30	38	0
3	33	39	1	20	34	43	0	37	30	40	1
4	32	37	0	21	32	39	0	38	35	41	1
5	32	43	1	22	36	42	1	39	32	42	1
6	33	42	0	23	33	37	0	40	33	40	0
7	31	45	1	24	30	38	0	41	34	43	0
8	36	45	1	25	36	43	1	42	32	38	0
9	31	34	0	26	33	41	0	43	34	42	1
10	32	37	1	27	32	39	0	44	30	41	1
11	36	44	1	28	30	36	0	45	34	40	0
12	32	41	0	29	30	36	0	46	34	42	1
13	32	40	0	30	36	42	1	47	33	38	0
14	33	35	0	31	33	38	0	48	34	44	1
15	31	42	1	32	33	38	0	49	35	43	0
16	34	41	0	33	35	41	1	50	35	44	1
17	35	37	0	34	35	41	1				



TERIMA KASIH