

# BIG DATA HAPPINESS

**MBC** LAB LEARNING ADVANCE SKILLS 2024

**WEEK 1**

# **INTRODUCTION TO DATA AND MACHINE LEARNING**

**MENTOR : MEI**

# Apa itu DATA??

**kumpulan informasi yang dapat diolah untuk mendapatkan wawasan atau pengetahuan yang bermanfaat**

**data adalah bahan baku yang digunakan untuk membangun model yang dapat memprediksi atau mengklasifikasikan informasi baru.**

<https://igracias.telkomuniversity.ac.id/index.php?pageid=2941>

psi-jogja-apr-2021.csv (35.98 kB)

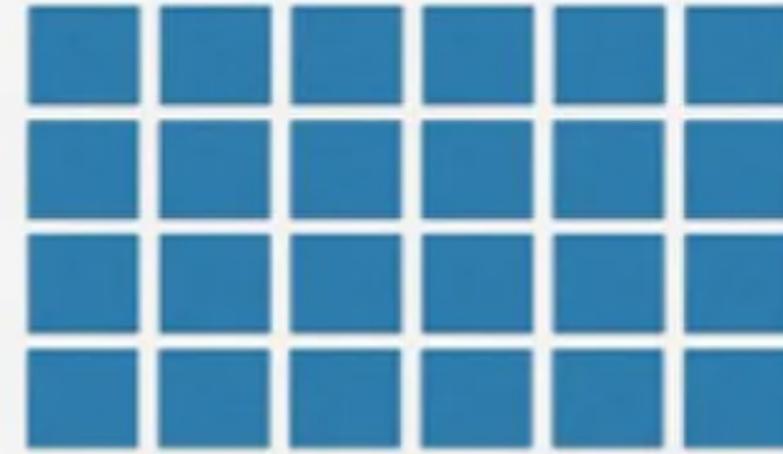
Detail Compact Column 10 of 11 columns ▾

Date	Time	# PM2.5	# PM10	# SO2	# CO	# O3	# NO2	# Max
4/1/2021	00:00:00	45	19	21	15	8	3	21
4/1/2021	01:00:00	44	18	20	14	8	3	20
4/1/2021	02:00:00	43	17	20	14	7	3	20
4/1/2021	03:00:00	40	17	20	13	7	3	20
4/1/2021	04:00:00	38	16	19	12	7	3	19
4/1/2021	05:00:00	35	14	19	12	7	3	19
4/1/2021	06:00:00	34	14	19	12	7	3	19
4/1/2021	07:00:00	33	13	19	11	7	3	19
4/1/2021	08:00:00	31	13	19	11	7	3	19
4/1/2021	09:00:00	31	13	19	11	7	3	19
4/1/2021	10:00:00	31	12	19	11	7	3	19
4/1/2021	11:00:00	31	12	19	11	8	3	19
4/1/2021	12:00:00	31	12	19	11	8	3	19
4/1/2021	13:00:00	32	13	19	11	8	3	19
4/1/2021	14:00:00	32	13	19	11	7	3	19
4/1/2021	15:00:00	33	13	19	11	7	3	19

# Jenis Jenis Data

## Structured

Data yang terorganisir dengan baik dalam format tabel dengan baris dan kolom, di mana setiap kolom memiliki tipe data yang spesifik.



What you find in a DB  
(typically)

- Mudah disimpan, diakses, dan dianalisis menggunakan SQL dan alat analisis data lainnya.
- Data memiliki skema yang tetap dan konsisten.

## Unstructured

Data yang tidak memiliki format atau struktur yang jelas, sehingga sulit untuk diatur dalam tabel relasional.



What you find in the 'wild'  
(text, images, audio, video)

- Membutuhkan teknik khusus untuk analisis, seperti Natural Language Processing (NLP) untuk teks atau pengenalan gambar (image recognition) untuk gambar.

# Format Data



**Comma-Separated Values adalah format file teks yang digunakan untuk menyimpan data dalam bentuk tabel. Setiap baris dalam file adalah rekaman data, dan setiap kolom dipisahkan oleh koma.**

```
Name, Age, Country  
Alice, 30, USA  
Bob, 25, Canada
```



**JavaScript Object Notation adalah format file yang digunakan untuk menyimpan dan bertukar data dalam bentuk pasangan kunci-nilai. JSON adalah format yang mudah dibaca manusia dan diproses oleh mesin.**

```
{  
  "Name": "Alice",  
  "Age": 30,  
  "Country": "USA"  
}
```

- CSV lebih sederhana dan cocok untuk data tabular yang terstruktur, sementara JSON lebih fleksibel dan dapat menyimpan data yang lebih kompleks.
- CSV lebih hemat ruang untuk data sederhana, sedangkan JSON menyediakan lebih banyak konteks dan struktur.

# Machine Learning Flow

- Understand Data Distribution
- Feature Collection
- Initial Hypotheses

EDA

01.

- Feature Scaling
- Encoding Categorical Variables
- Sampling

Data  
Preprocessing

02.

Data  
Cleansing

- Handling Missing Values
- Outlier Detection
- Correcting Inconsistencies

- Model Performance Metrics

Evaluation

05.

Modelling

- Algorithm Selection
- Training the Model
- Testing the Model
- Hyperparameter Tuning

Deploy

- Model Integration

06.

# Apa itu MACHINE LEARNING?

Machine adalah proses dimana komputer belajar membuat keputusan dari data tanpa diprogram secara eksplisit.

Example :

Predict Notification



OR

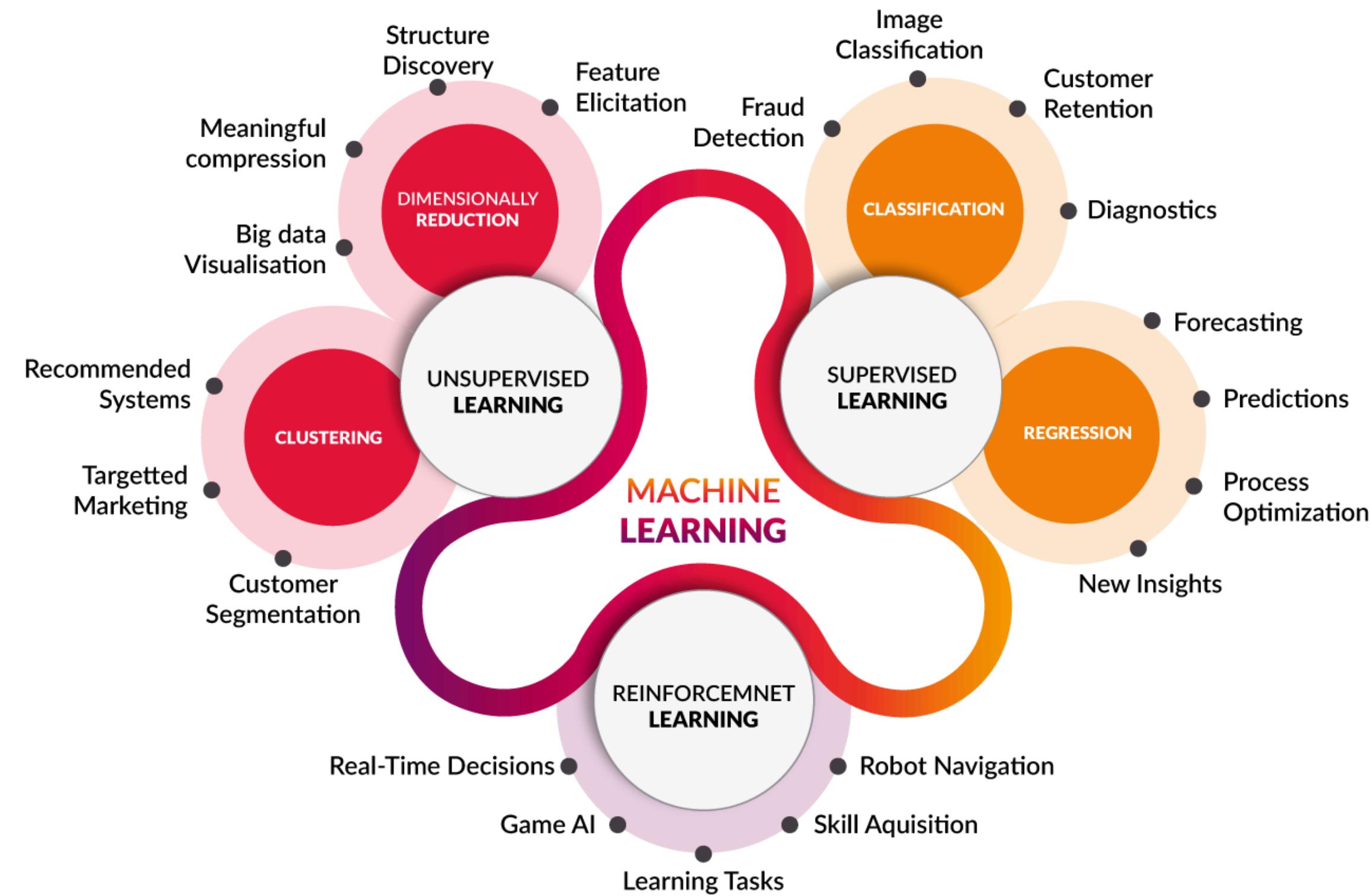


Based on Content or Sender



Cluster Book based on words  
they contain

# JENIS MACHINE LEARNING



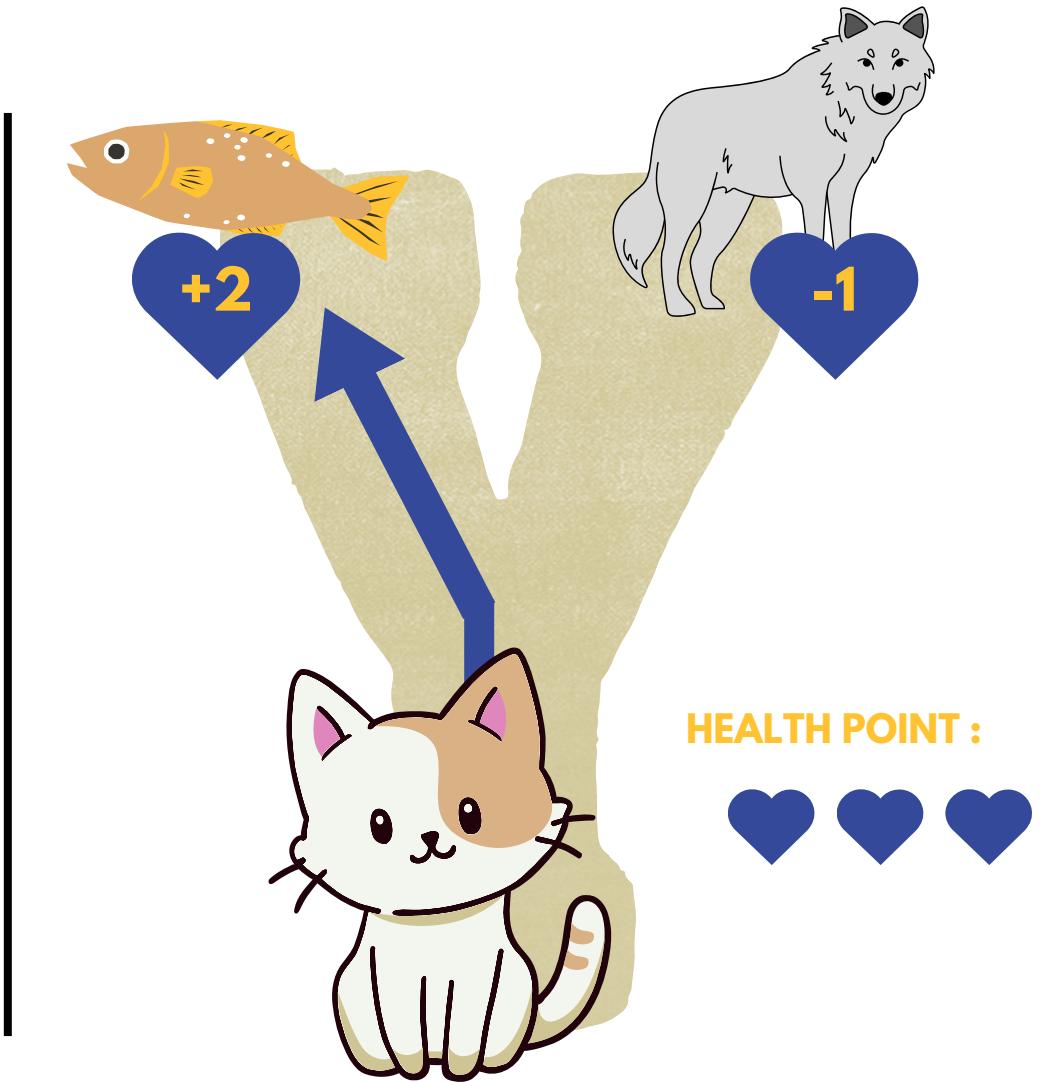
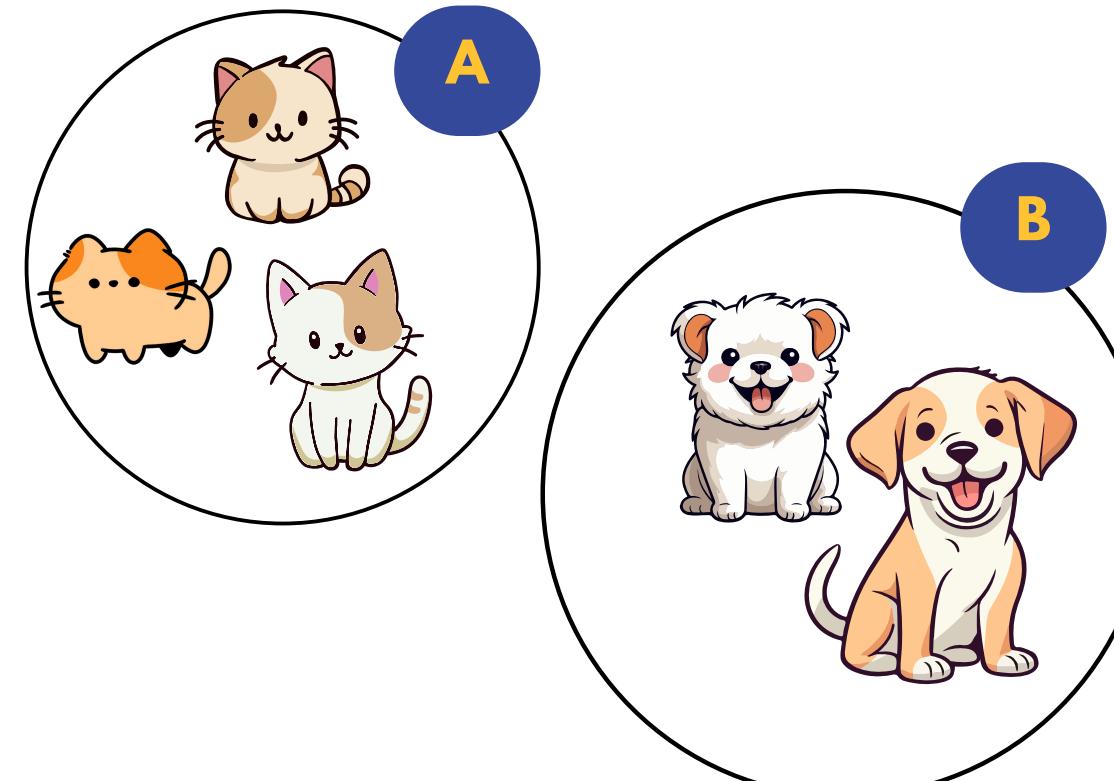
# JENIS MACHINE LEARNING



**SUPERVISED  
LEARNING**



**UNSUPERVISED  
LEARNING**



**REINFORCEMENT  
LEARNING**

**WEEK 2**

# **EXPLORATORY DATA ANALYSIS AND DATA VISUALIZATION**

**MENTOR : AGN**

# EXPLORATORY DATA ANALYSIS

Tujuan utama EDA adalah untuk membantu melihat data sebelum membuat asumsi apa pun

EDA mudah karena data nya sedikit dan kolomnya kurang bervariasi dari segi value maupun arti dari tiap kolomnya

# EXPLORATORY DATA ANALYSIS & DATA VISUALIZATION

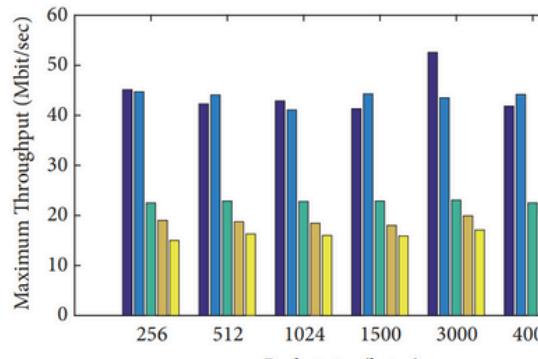
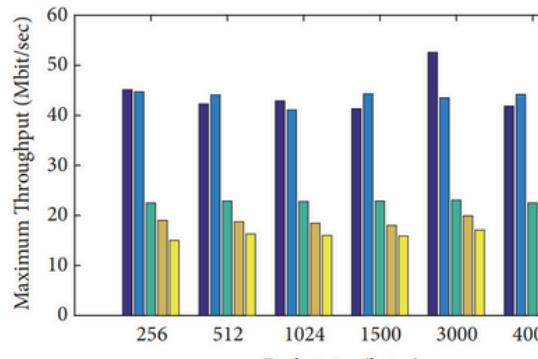
## Why EDA so Important ?

2020 3rd International Conference on Hot Information-Centric Networking

### The Impact of Chunk Size on Named Data Networking Performance

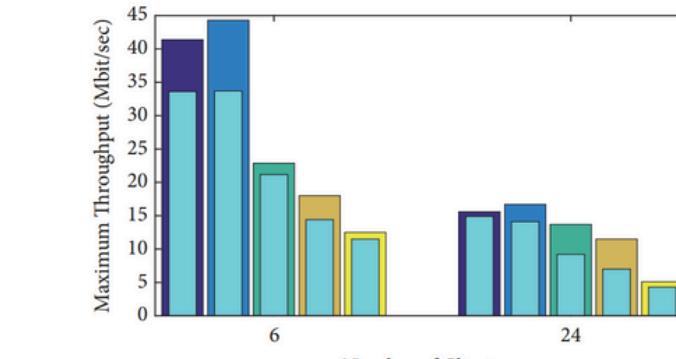


### Research Article On the Tradeoff between Performance and Programmability for Software Defined WiFi Networks



■ Netgear  
■ Netgear-Openwrt  
■ Odin-V2-1App

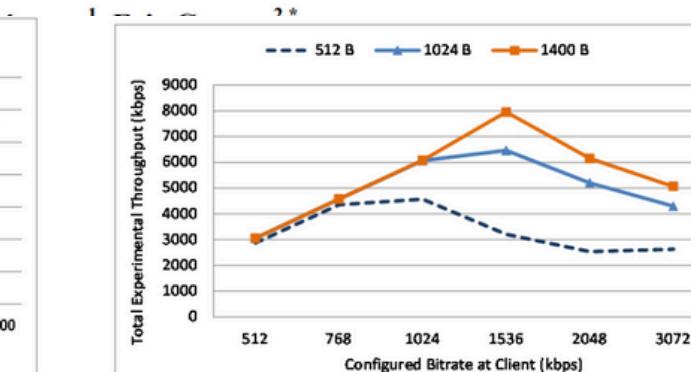
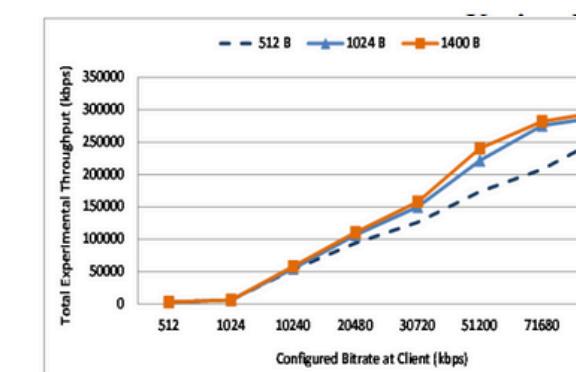
■ Odin-V2-2App  
■ Odin-V2-4App



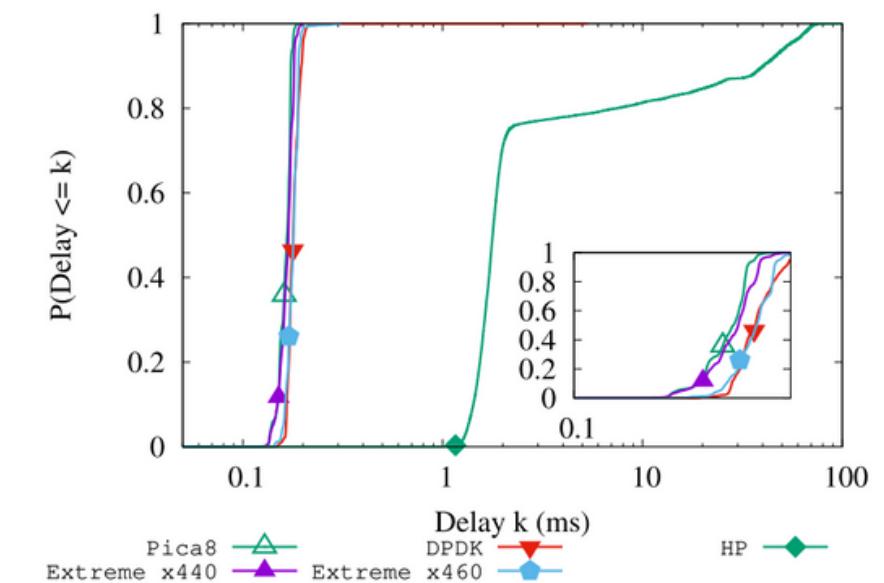
■ Netgear  
■ Netgear-Openwrt  
■ Odin-V2-1App

■ Odin-V2-2App  
■ Odin-V2-4App

### Network Performance Evaluation Based on Three Processes



OpenFlow data planes performance evaluation



■ Pica8  
▲ Extreme x440  
▼ Extreme x460  
◆ HP

# Data Log Router NDN

## Consumer

- Skenario 1
- Skenario 2
- Skenario 3
  - Cpu
    - cpu-0
      - percent-idle-2024-03-04
        - epoch
        - value
      - percent-interrupt-2024-03-04
      - percent
      - dll
  - Log
  - performance
    - cpu-1
    - cpu-2
    - cpu-3
    - load
    - memory
    - thermal cooling\_device

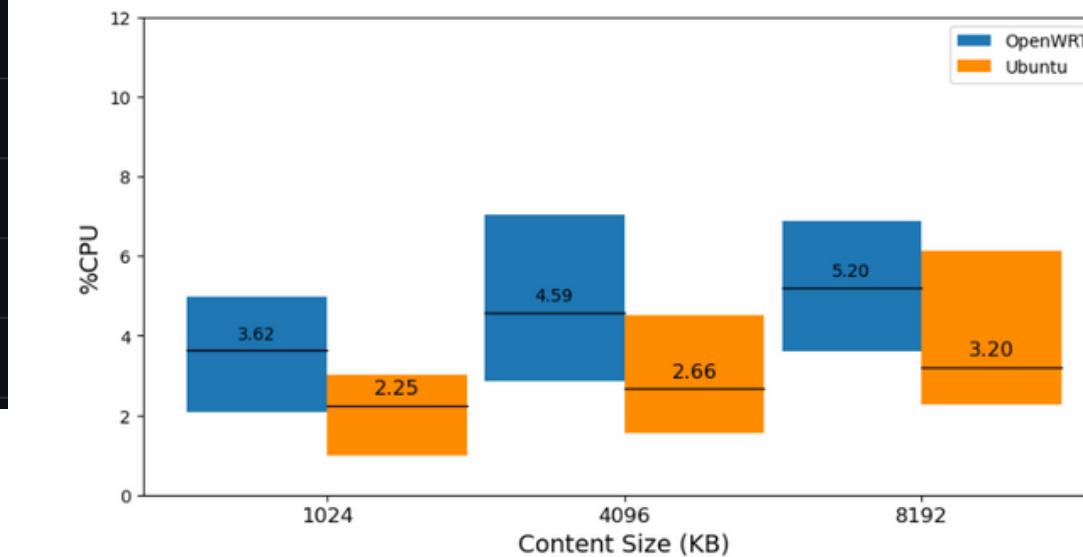
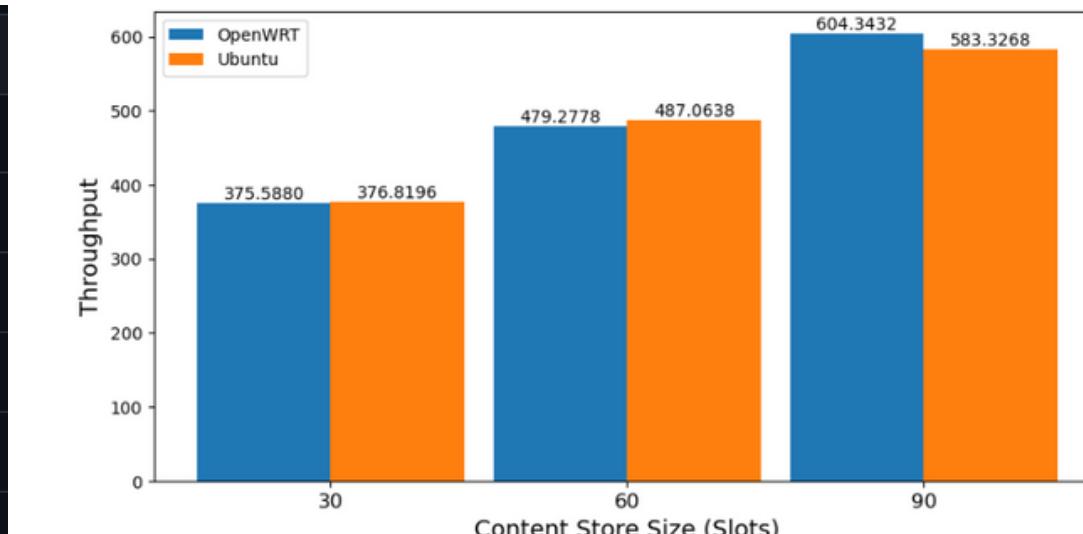
## Producer

percent-idle-2024-03-04.csv
percent-interrupt-2024-03-04.csv
percent-nice-2024-03-04.csv
percent-softirq-2024-03-04.csv
percent-steal-2024-03-04.csv
percent-system-2024-03-04.csv
percent-user-2024-03-04.csv
percent-wait-2024-03-04.csv

epoch	value
1709537591.383	94.897959
1709537592.383	94.897959
1709537593.383	99.000000

Name
..
cpu-0
cpu-1
cpu-2
cpu-3
load
memory
thermal-cooling_device0
thermal-cooling_device1
thermal-thermal_zone0
thermal-thermal_zone1

## Output



Based on Our Research about NDN Performance

**400 Files, 3000 Rows**

# CONTOH SIMPLE

Anda bekerja sebagai data analyst di sebuah restoran dan diminta untuk melakukan analisis terhadap data penjualan mereka. Restoran ingin memahami pola penjualan harian dan mengidentifikasi hari-hari terbaik untuk promosi. Anda diberikan data penjualan selama satu bulan.

- **Date:** Tanggal penjualan
- **Day of Week:** Hari dalam seminggu (Senin, Selasa, dll.)
- **Total Sales:** Total nilai penjualan hari tersebut
- **Number of Transactions:** Jumlah transaksi yang terjadi
- **Average Transaction Value:** Nilai rata-rata per transaksi

# Kesimpulan Contoh simple

- Berdasarkan analisis Anda, hari apa yang paling menguntungkan untuk penjualan?
- Apakah ada hari tertentu dalam seminggu yang menunjukkan penjualan yang konsisten lebih rendah? Jika ya, apa yang dapat direkomendasikan untuk meningkatkan penjualan pada hari-hari tersebut?

# Contoh yang tidak Simple

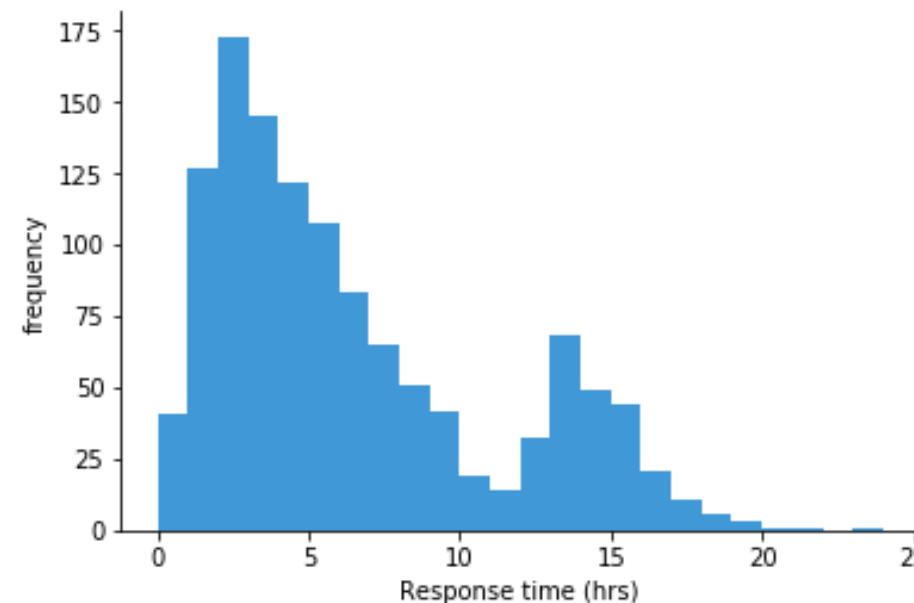
#	Column	Non-Null Count	Dtype
0	Year	161 non-null	int64
1	Company	161 non-null	object
2	Category	161 non-null	object
3	Market Cap(in B USD)	160 non-null	float64
4	Revenue	161 non-null	float64
5	Gross Profit	161 non-null	float64
6	Net Income	161 non-null	float64
7	Earning Per Share	161 non-null	float64
8	EBITDA	161 non-null	float64
9	Share Holder Equity	161 non-null	float64
10	Cash Flow from Operating	161 non-null	float64
11	Cash Flow from Investing	161 non-null	float64
12	Cash Flow from Financial Activities	161 non-null	float64
13	Current Ratio	161 non-null	float64
14	Debt/Equity Ratio	161 non-null	float64
15	ROE	161 non-null	float64
16	ROA	161 non-null	float64
17	ROI	161 non-null	float64
18	Net Profit Margin	161 non-null	float64
19	Free Cash Flow per Share	161 non-null	float64
20	Return on Tangible Equity	161 non-null	float64
21	Number of Employees	161 non-null	int64
22	Inflation Rate(in US)	161 non-null	float64

# **EDA**

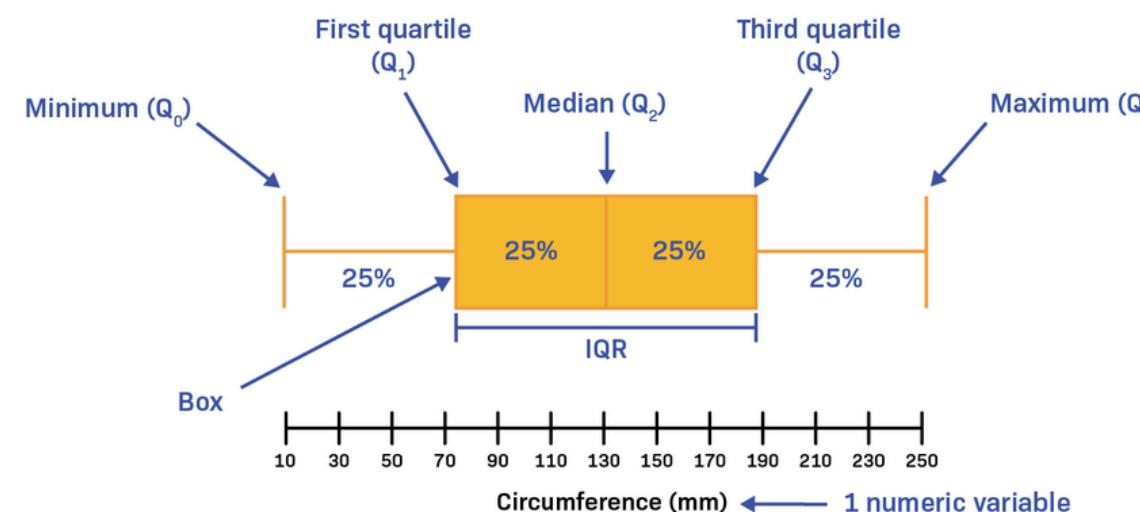
- **Analisis Univariat**
- **Analisis bivariat**
- **Analisis Multivariat**
- **Time Series Analysis**
- **Spatial Data Analysis**

# Analisis Univariat

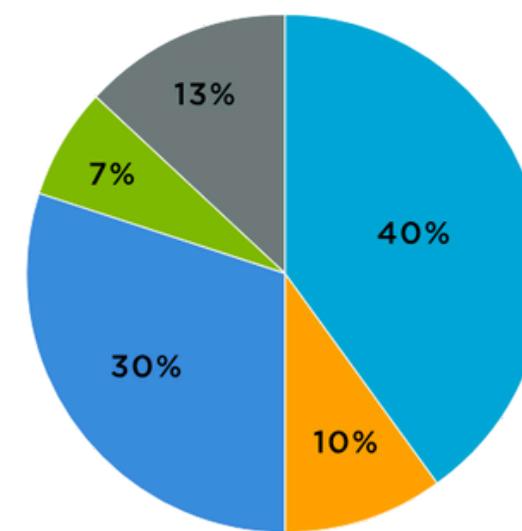
analisis ini berfokus pada satu variabel **tunggal**. Tujuannya adalah untuk menggambarkan dan merangkum data tersebut.



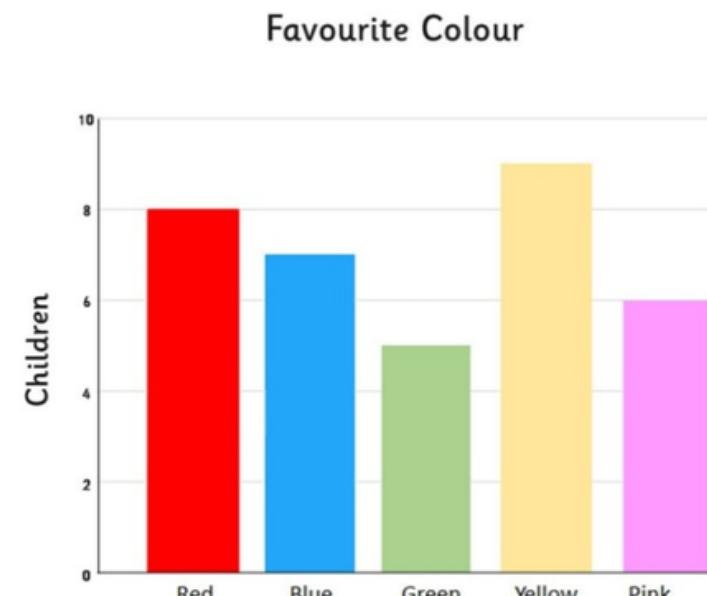
Histogram



Box Plot

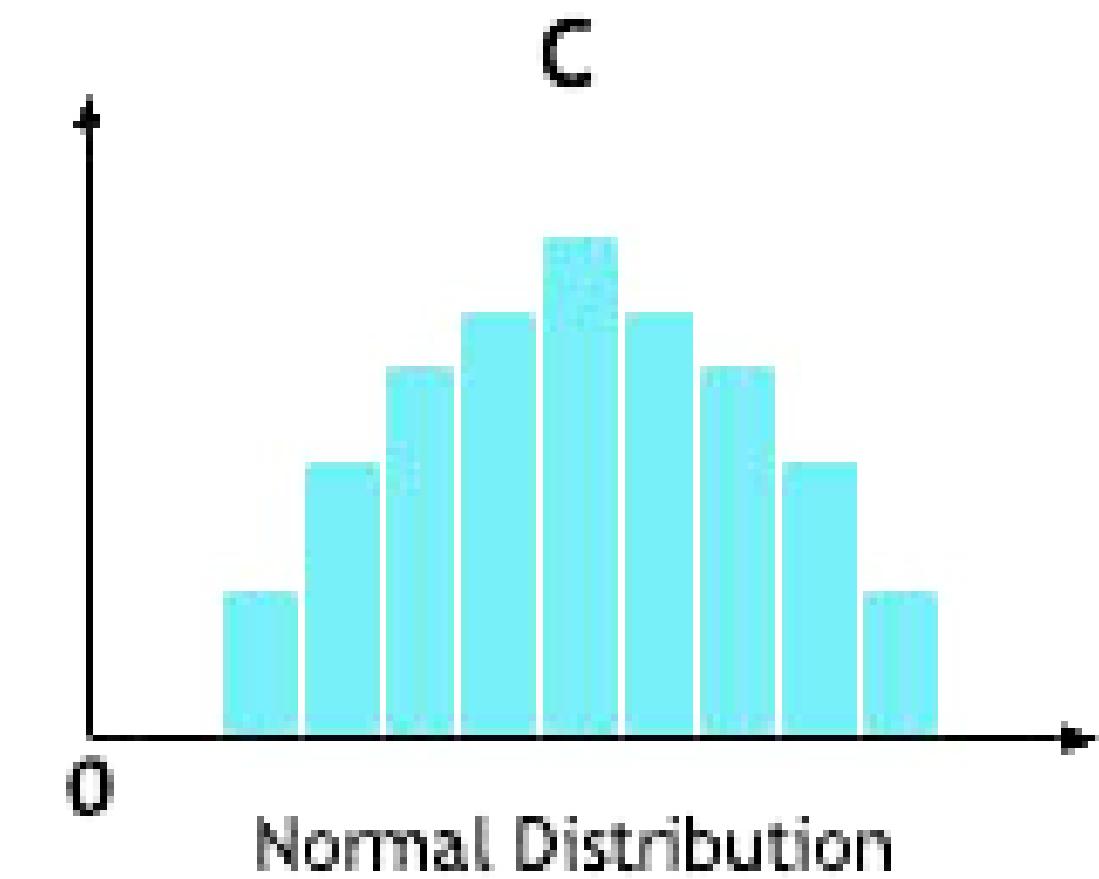
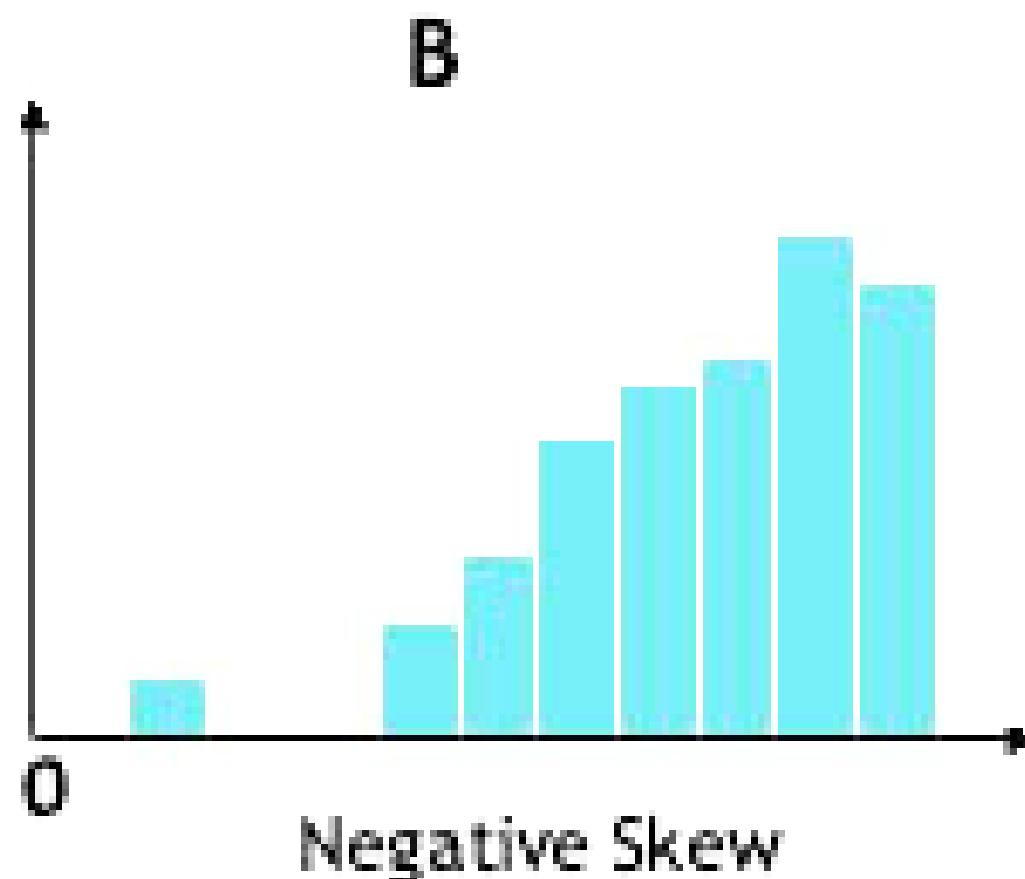
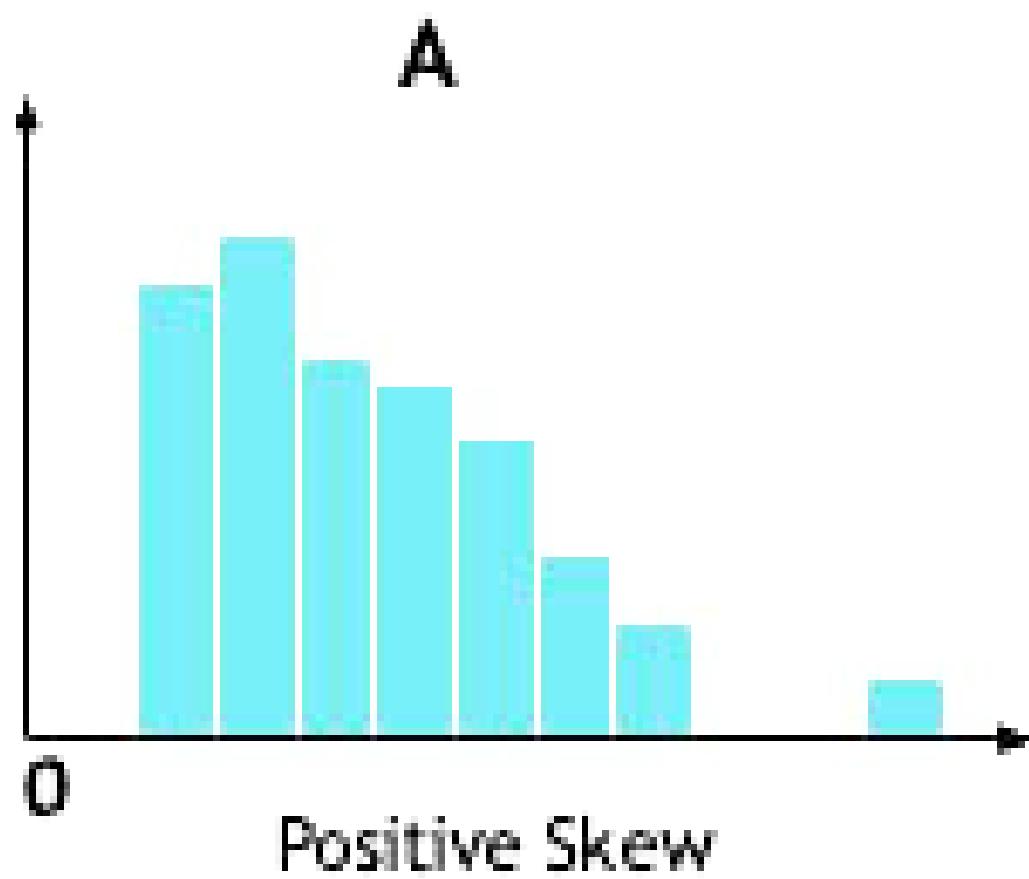


Pie Chart



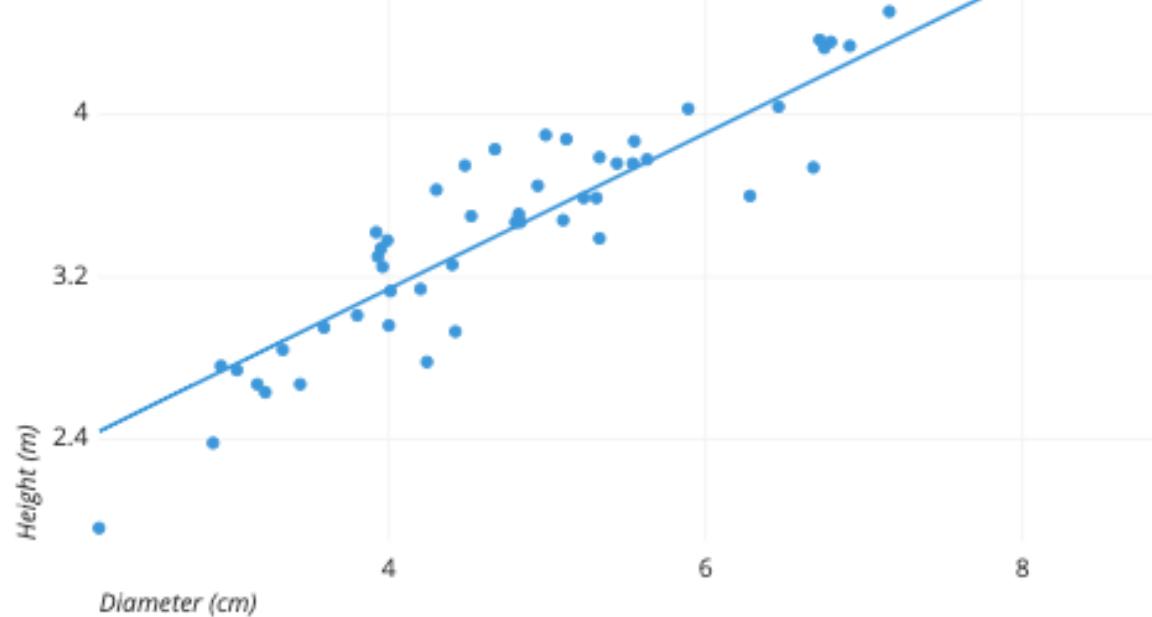
Bar Plot

# Histogram

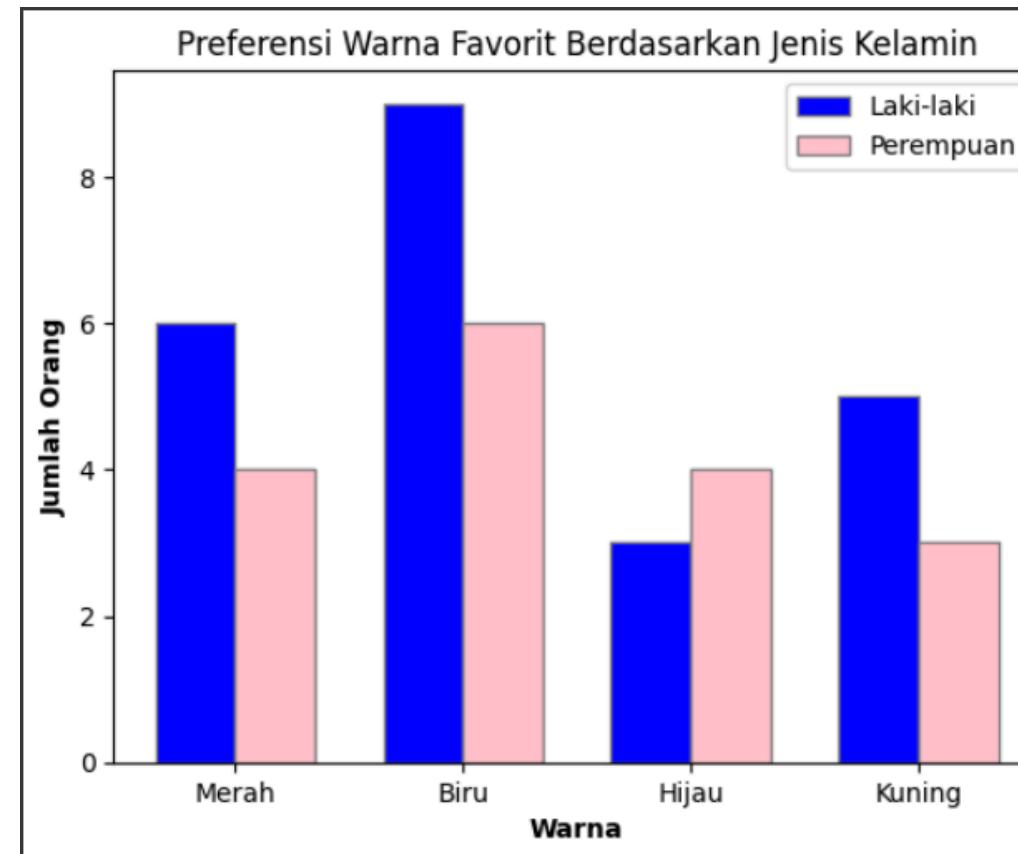


# Analisis Bivariat

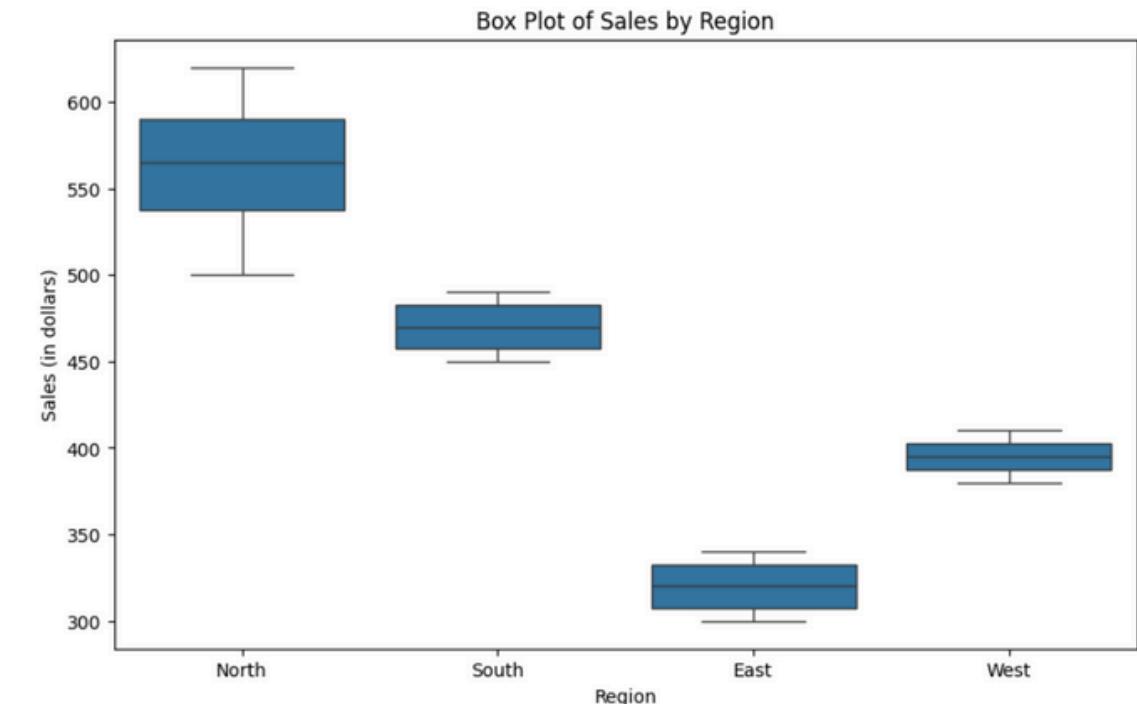
ini adalah proses menganalisis **dua variabel** untuk mengevaluasi hubungan sebab akibat, korelasi, dan ketergantungan di antara mereka.



Scatter Plot

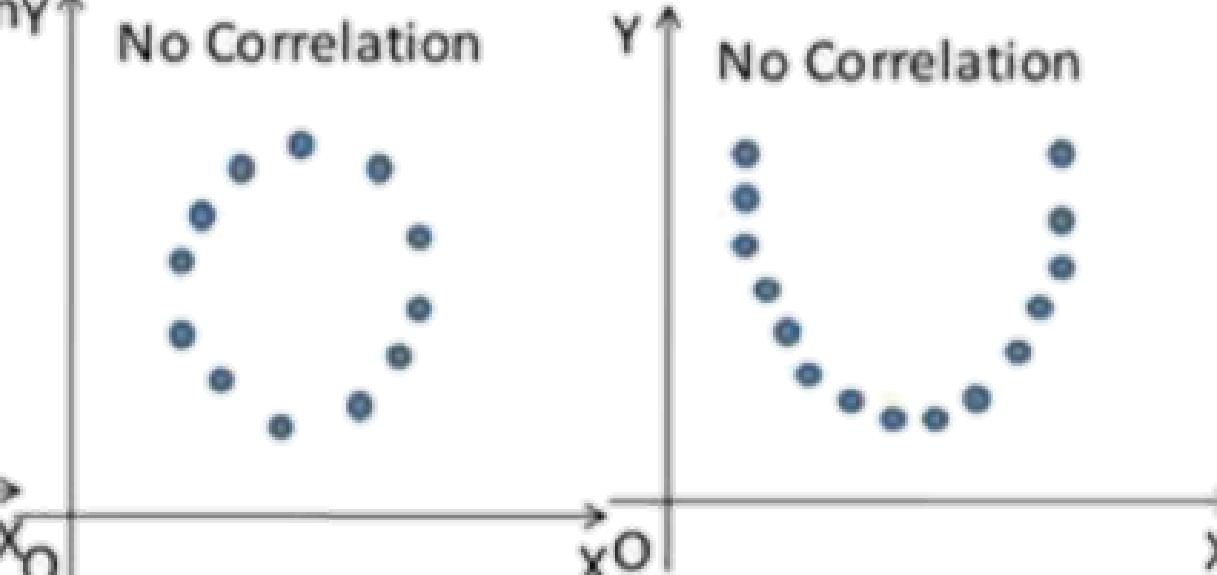
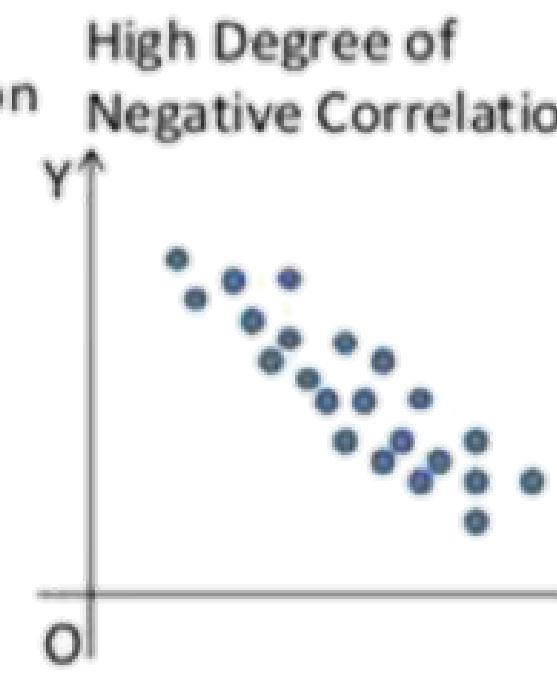
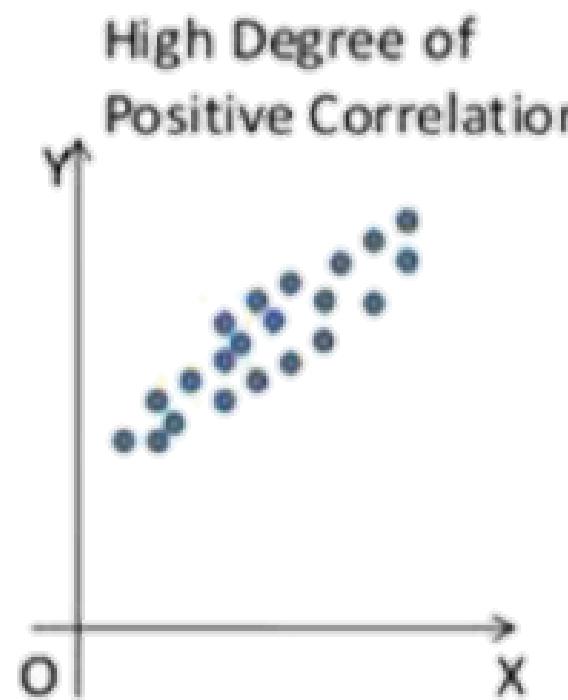
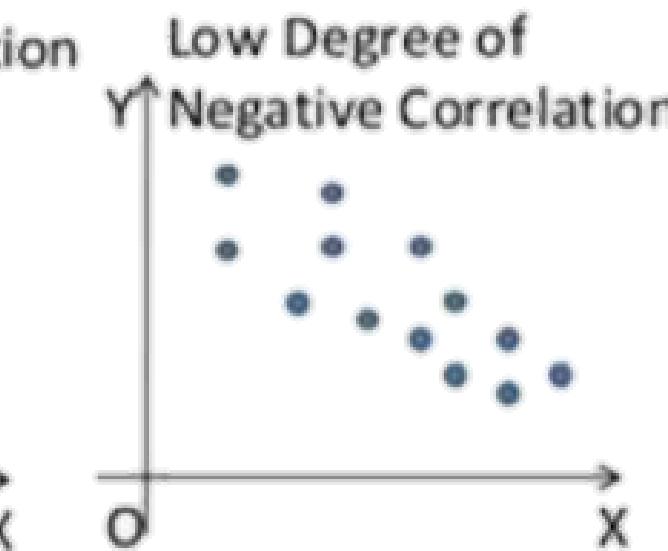
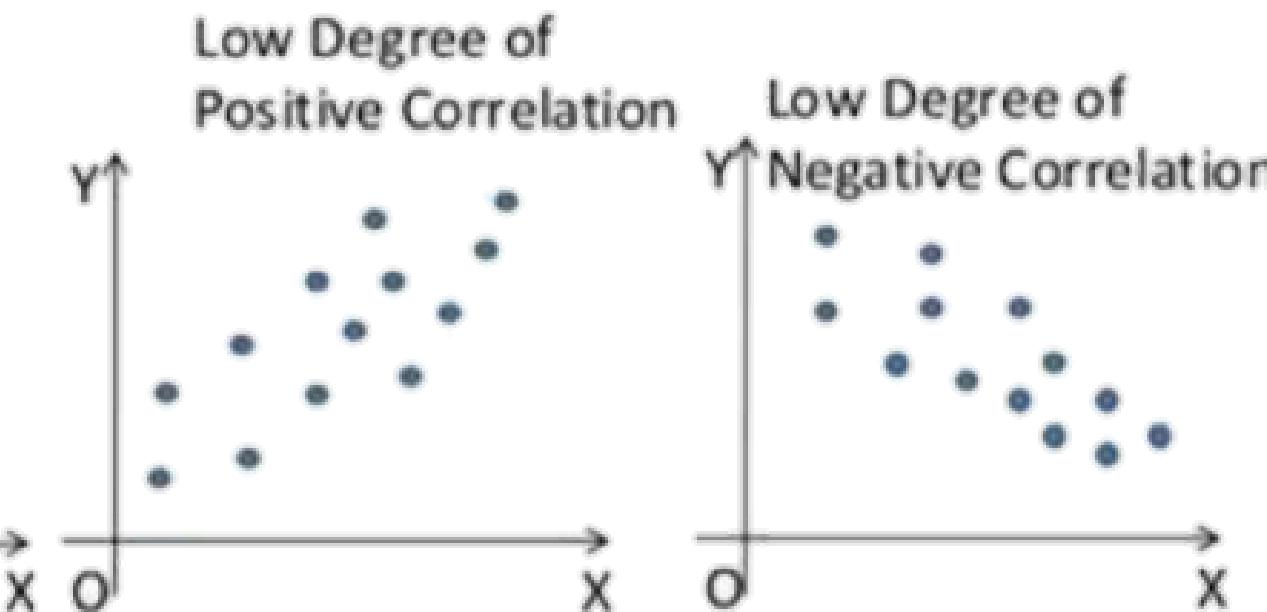
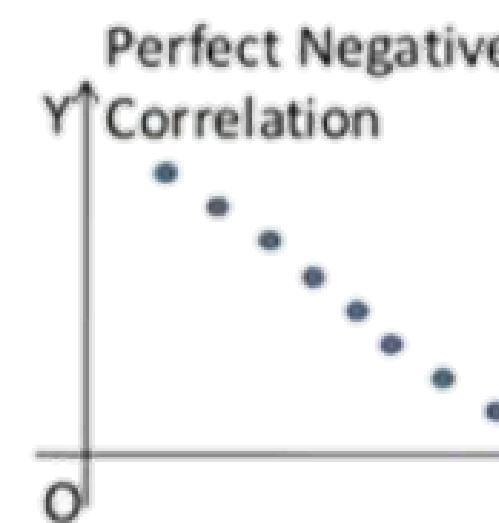
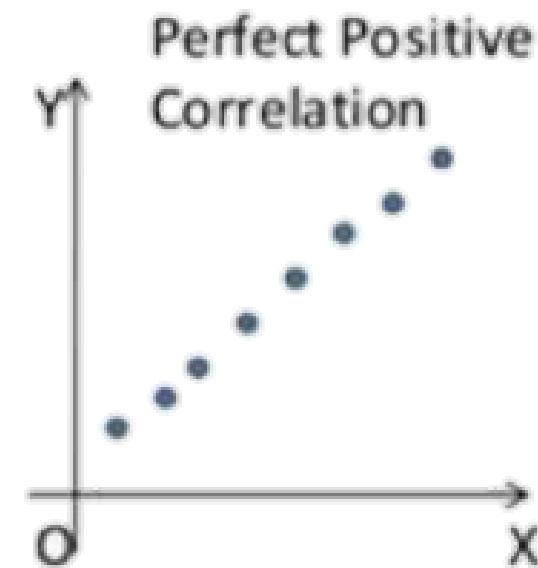


Bar Plot



Box Plot

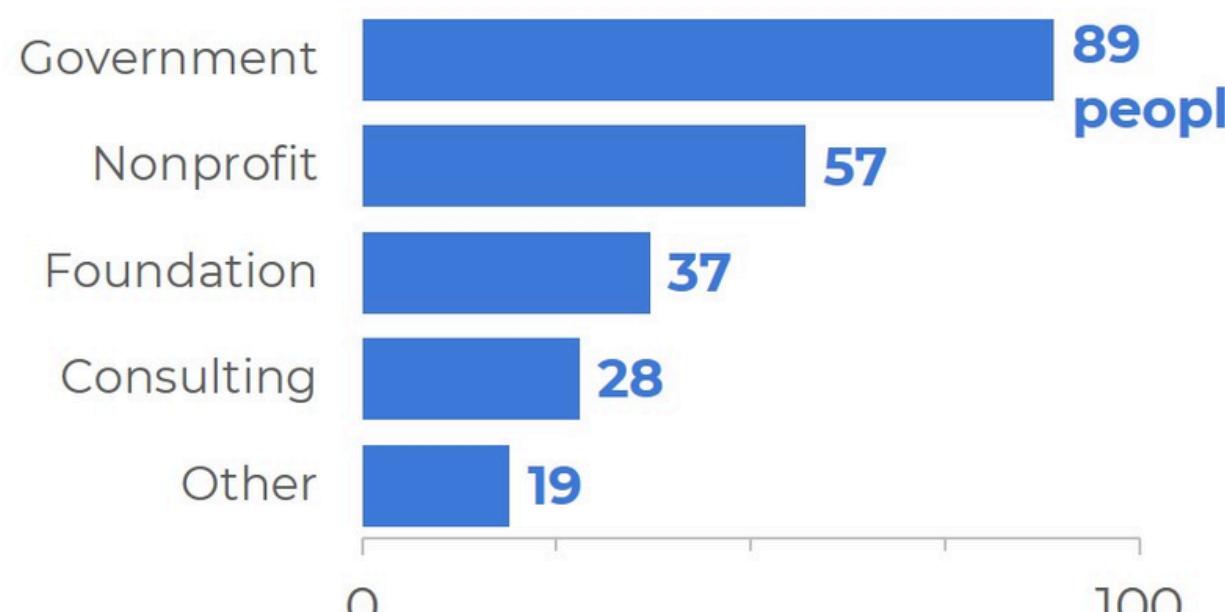
# Scatter Plot



# Bar Plot

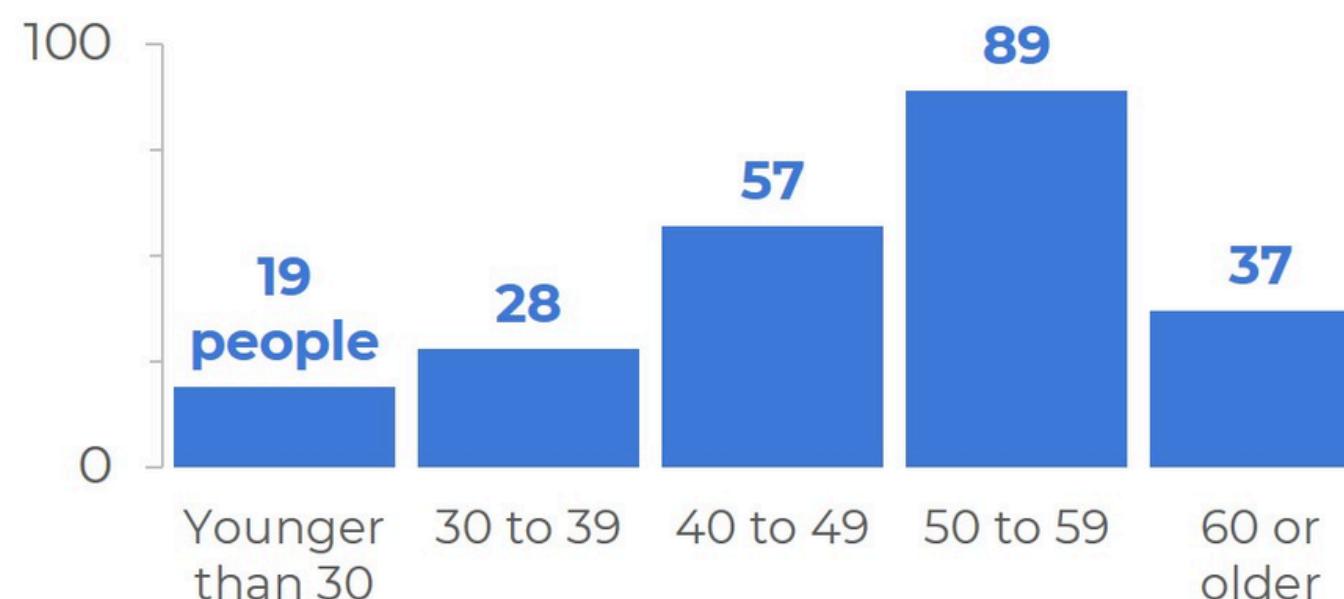
## Horizontal

Nominal/categorical



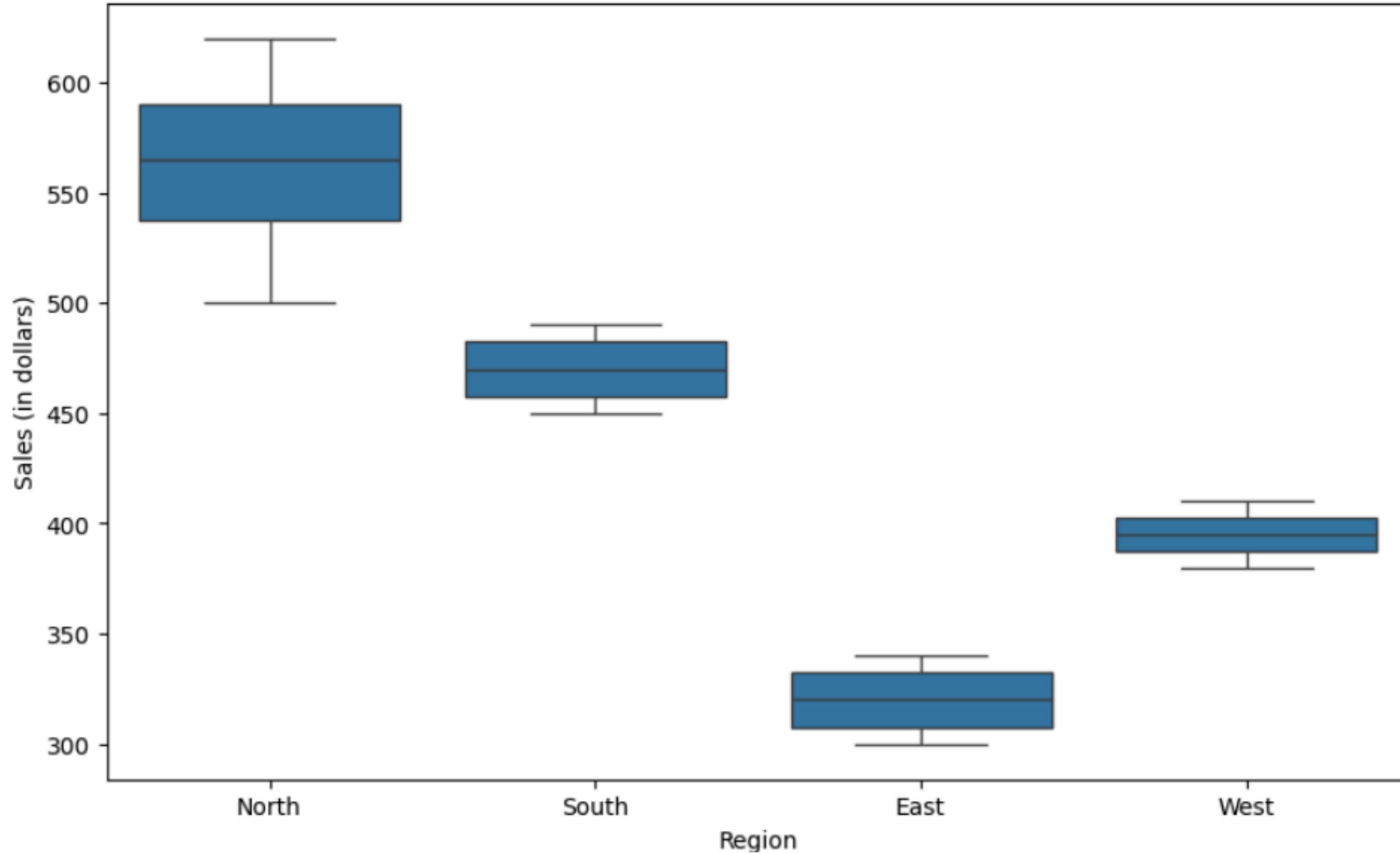
## Vertical

Ordinal/sequential



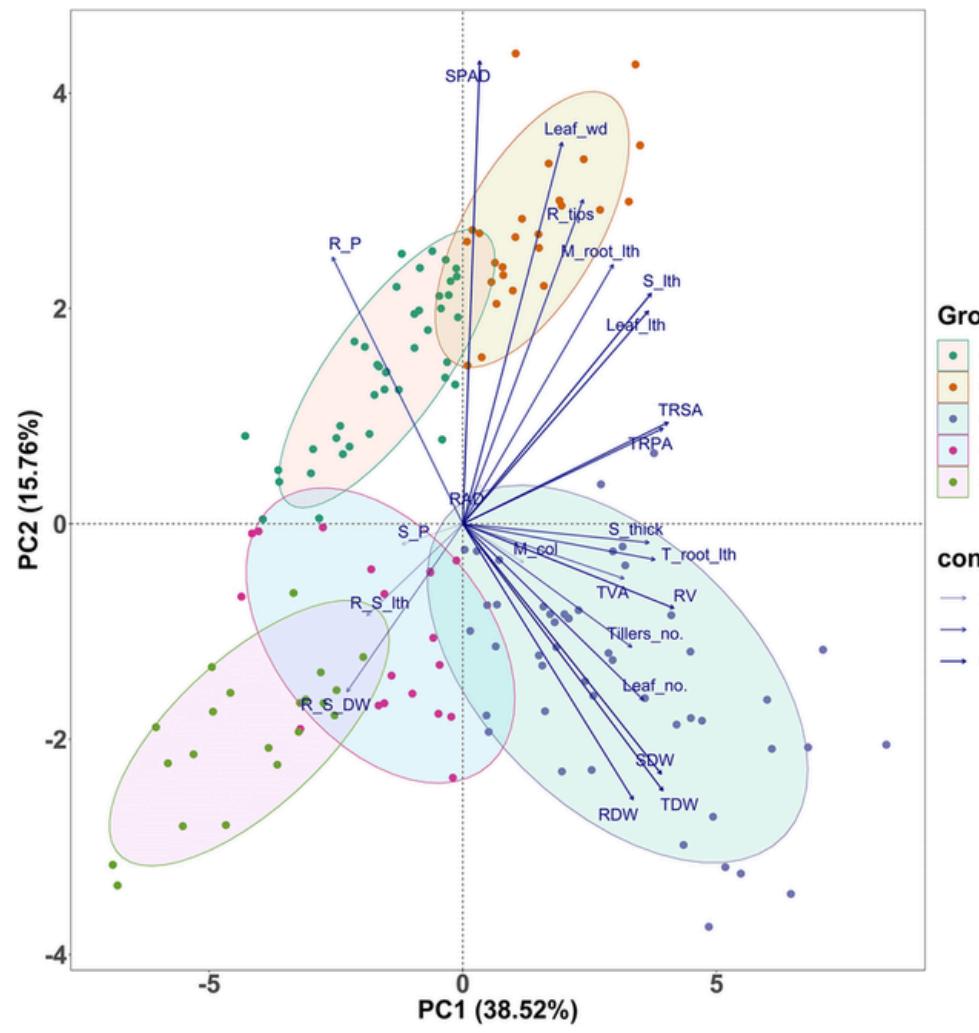
# Box Plot

Box Plot of Sales by Region



# Analisis Multivariat

teknik ini melibatkan analisis simultan lebih dari dua variabel untuk memahami hubungan antara lebih dari dua variabel secara bersamaan.



## PCA (Principal Componen Analysis)

Metode untuk mengurangi dimensi data sambil mempertahankan varians maksimum

**biplot**

Korelasi antara variabel dan komponen utama

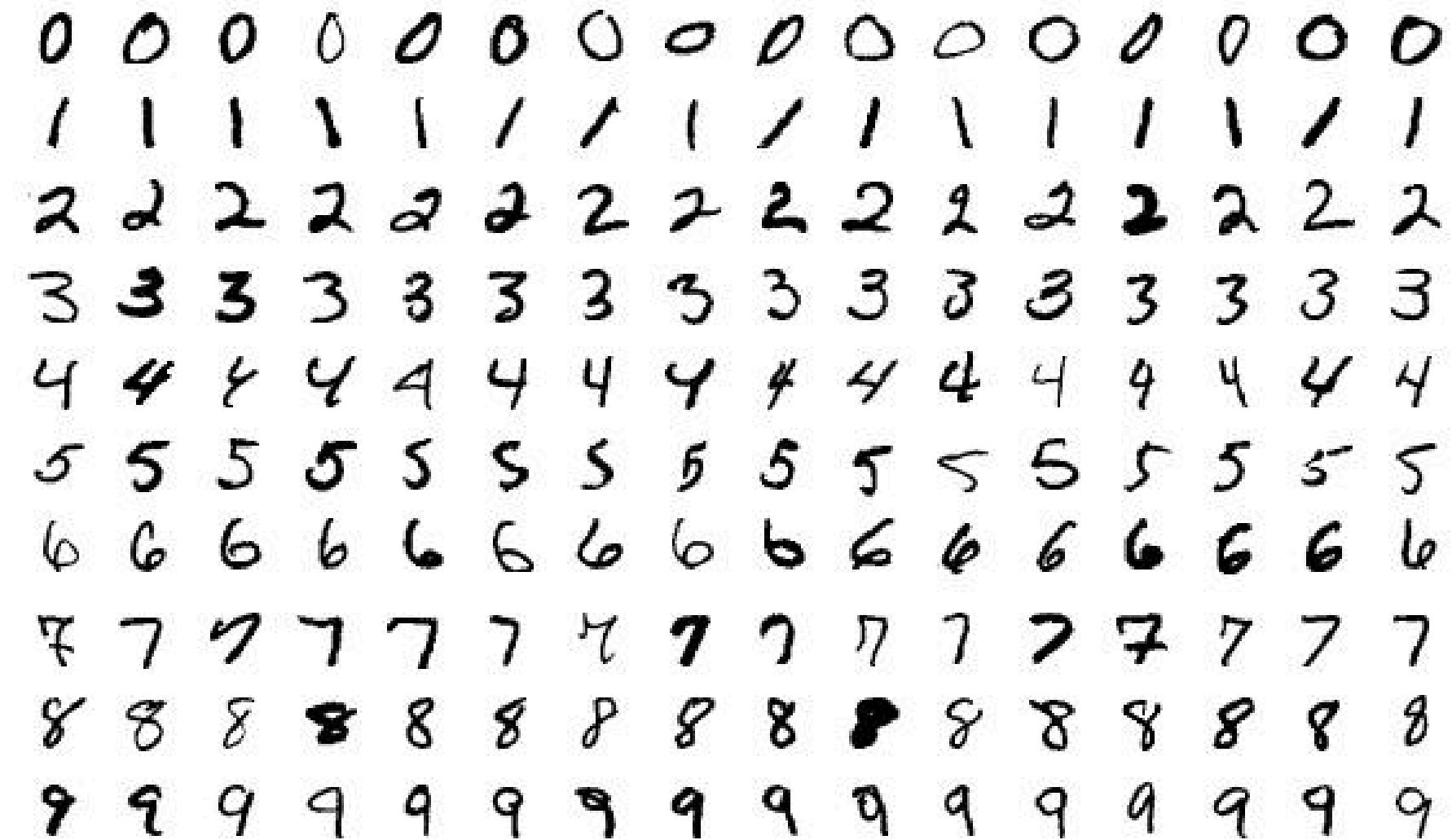
# PCA (Principal Component Analysis)

The image displays a 10x10 grid of square images, each representing a different object category from the ImageNet dataset. The categories are listed vertically on the left side of the grid:

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck

Each category has a row of 10 images. The images are arranged in a grid where each row contains 10 images and each column contains 10 images. The images are diverse within each category, showing various models, colors, and angles.

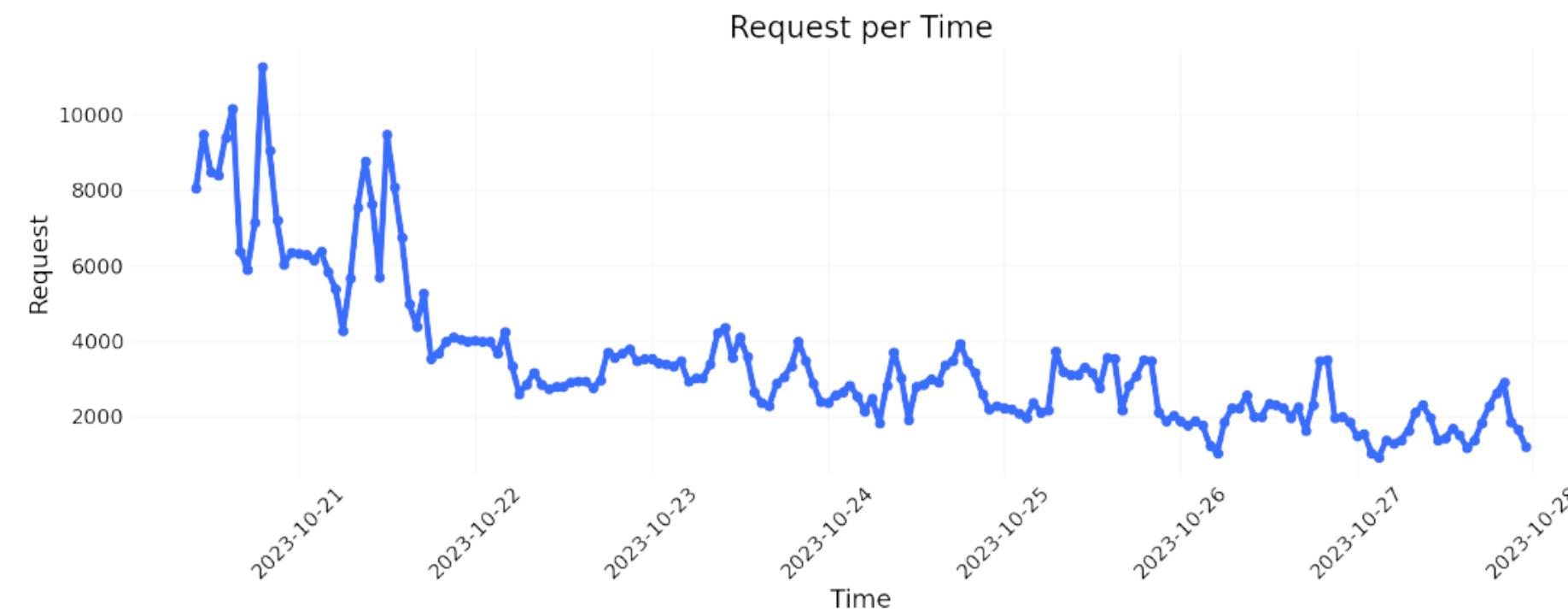
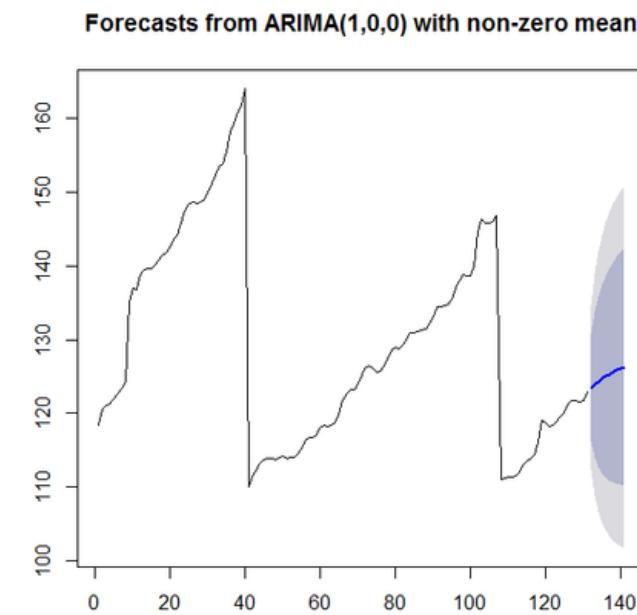
# Cifar 10



## MNIST

# Time Series Analysis

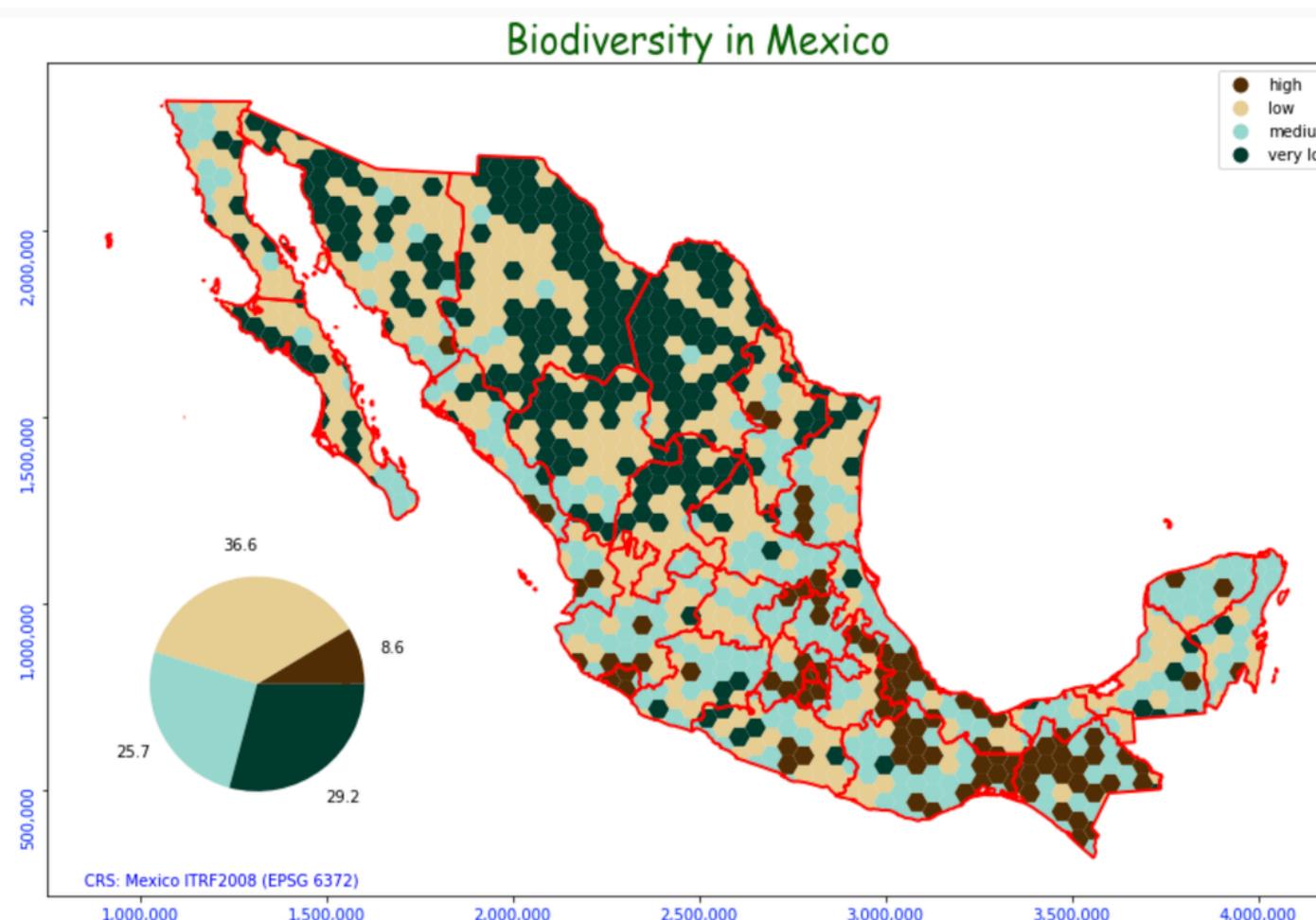
Metode analisis untuk data yang dikumpulkan secara berkala seiring waktu. Tujuannya adalah untuk mengidentifikasi tren dan pola musiman, dalam data.



**Line Plot**

# Spatial Data Analysis

Analisis data spasial adalah proses menganalisis data yang memiliki komponen geografis untuk memahami pola dan hubungan di ruang. Analisis ini membantu dalam pengambilan keputusan dengan memberikan wawasan tentang distribusi dan hubungan geografis dalam berbagai konteks, seperti perencanaan kota, manajemen lingkungan, dan penelitian kesehatan.



Tidak hanya Divisi GIS,  
Divisi Big Data juga  
harus memahami data  
spasial

# Do and Don't

## Do

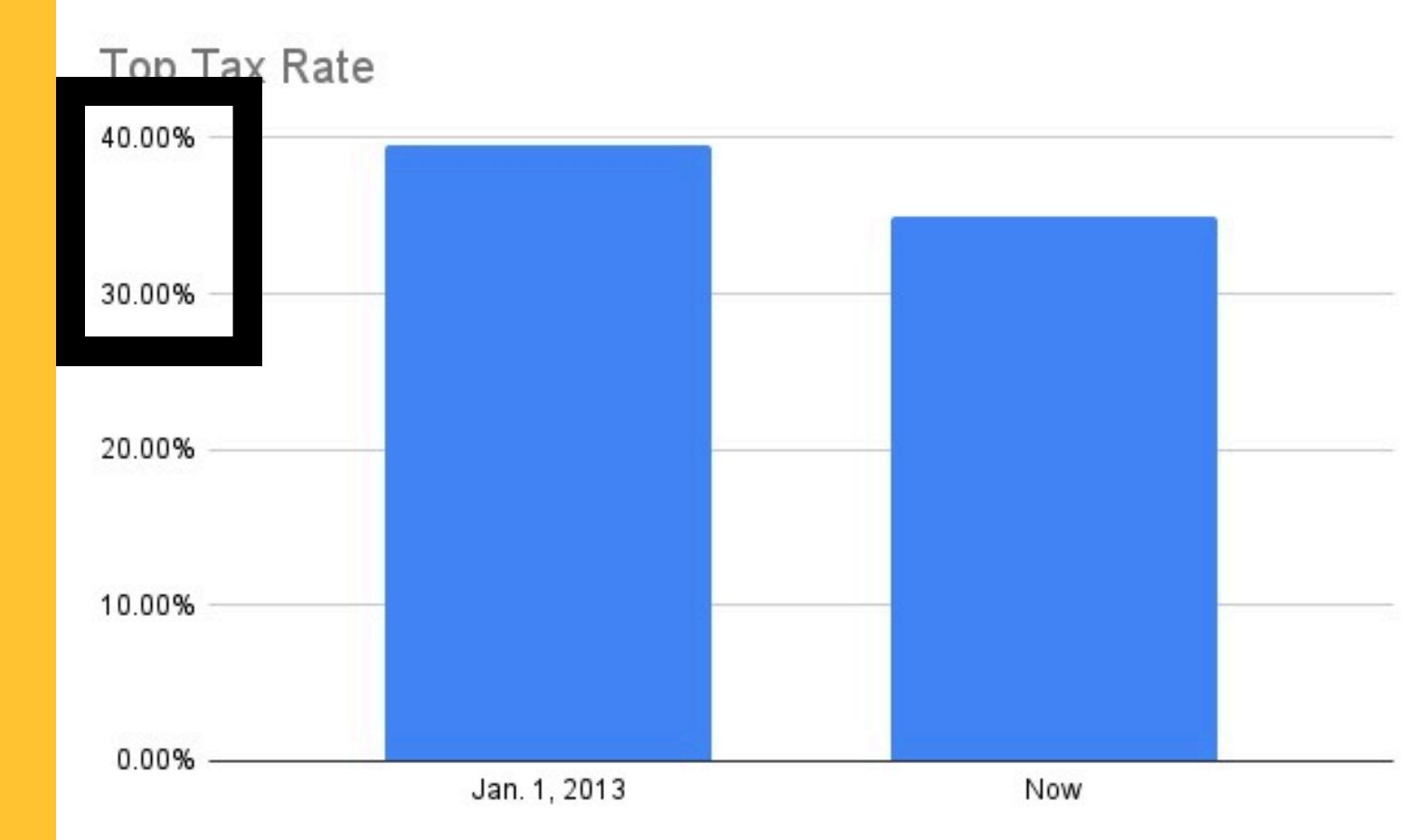
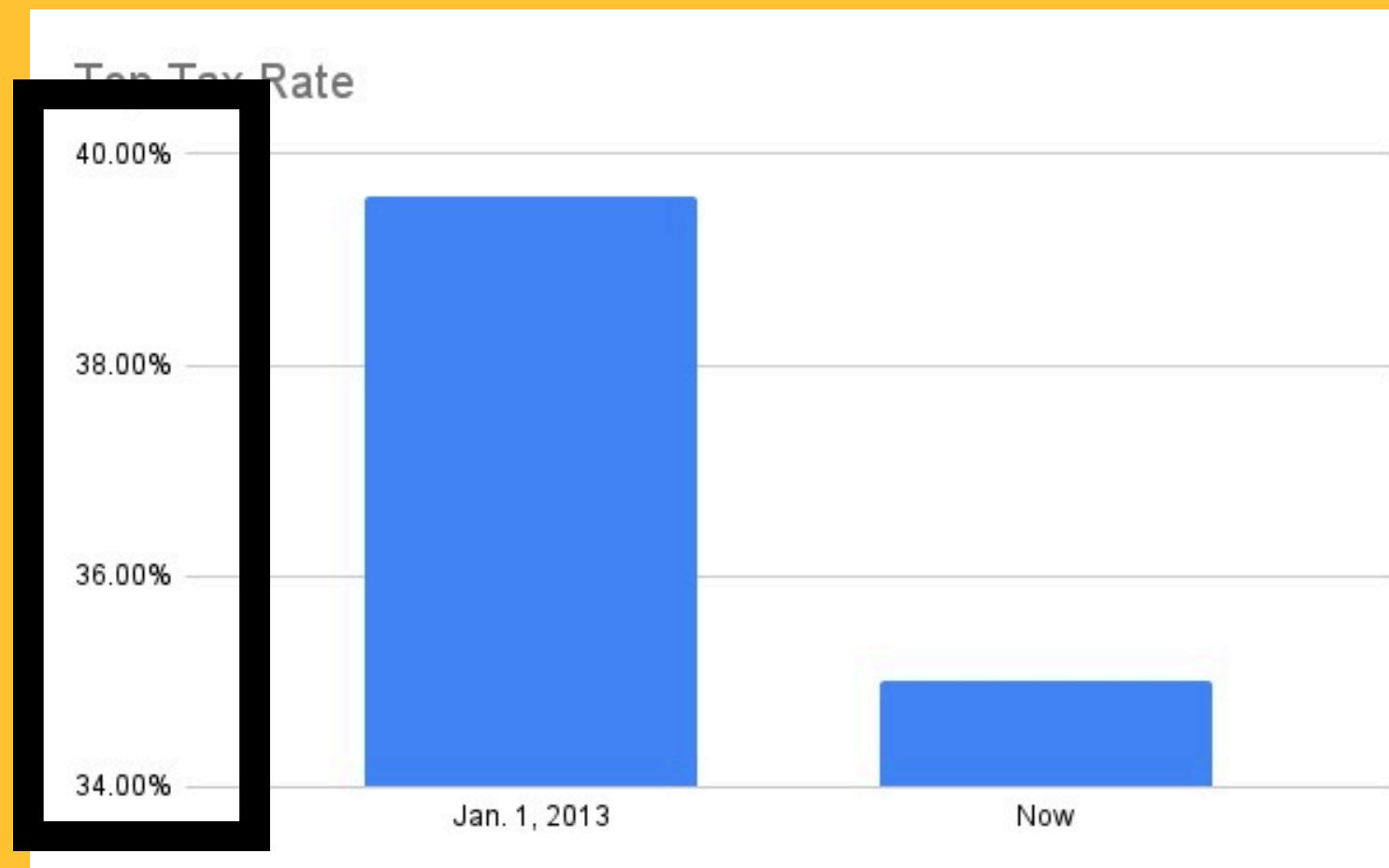
- Start Simple
- Gunakan Visualisasi yang Tepat
- Gunakan Warna yang Bijak
- Pahami lebih dalam mengenai Data

## Don't

- Kompleks
- Menggunakan Visualisasi data yang sama padaha data berbeda
- Tidak mengatur warna dan menyulitkan membedakan
- Tidak memahami kolom pada data dan langsung drop data

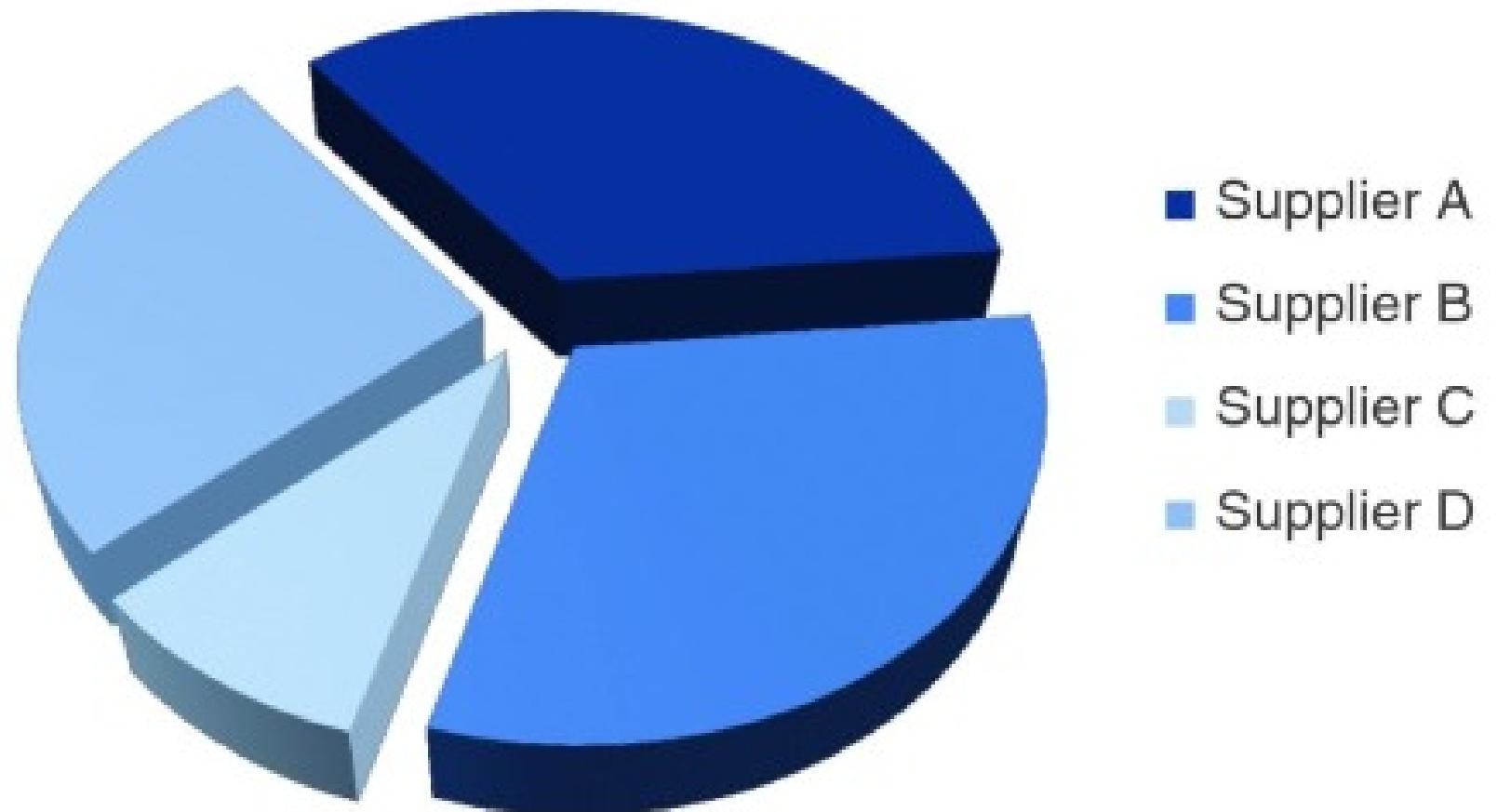
# Prinsip Visualisasi Data

## Misleading



# Prinsip Visualisasi Data

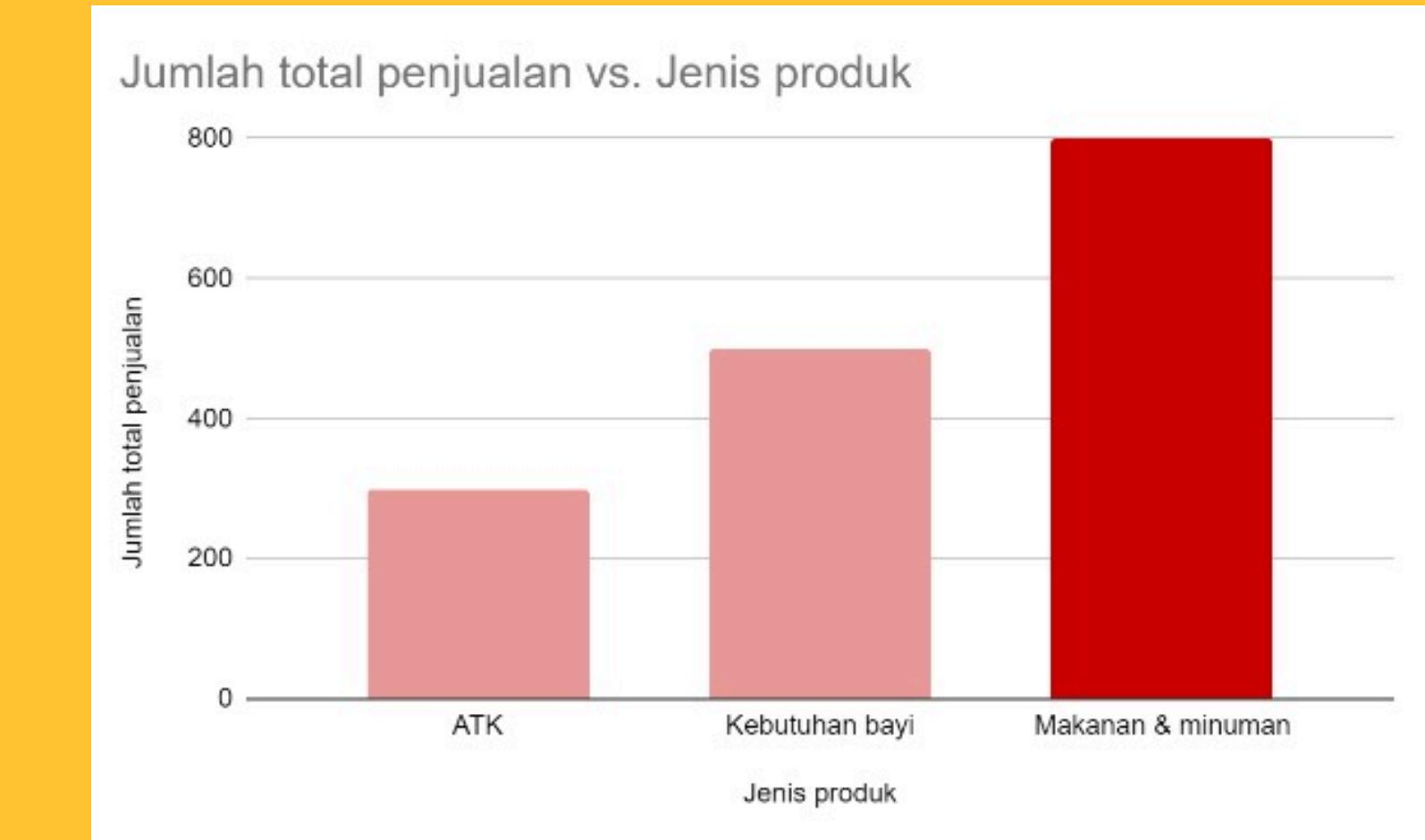
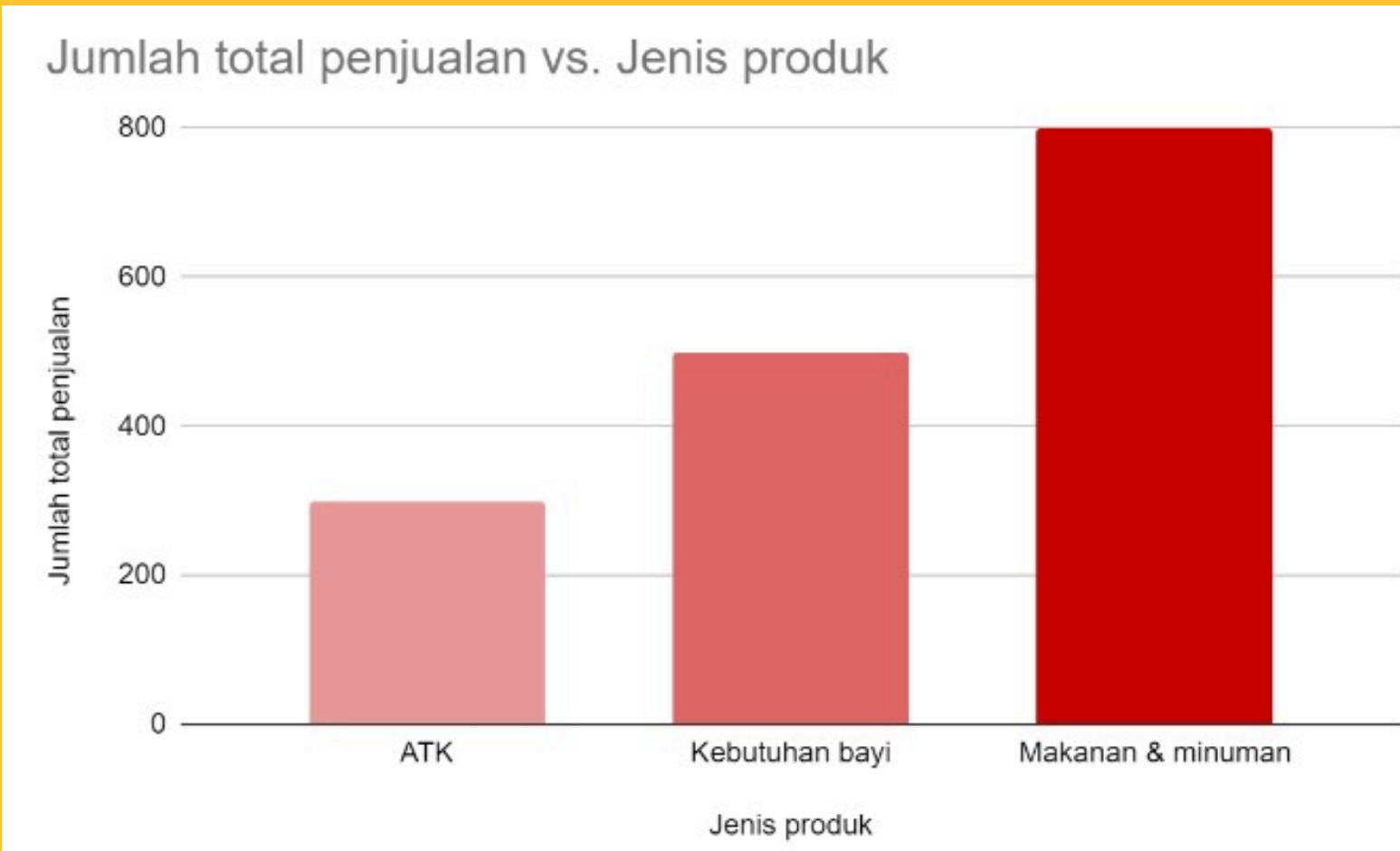
## Hiding



1. Berapa besar market share yang dimiliki supplier A?
2. Supplier manakah yang memiliki market share terbesar?

# Prinsip Visualisasi Data

## Distracts



**WEEK 3**

# **DATA CLEANSING AND DATA PREPROCESSING**

**MENTOR : JMD**

# DATA CLEANSING

Data cleansing adalah proses yang melibatkan persiapan data untuk memperbaiki atau menghapus kesalahan, menangani inkonsistensi, ketidakakuratan, dan bagian yang tidak relevan dari dataset.

Data yang bersih dan akurat sangat penting untuk melatih model ML, karena menggunakan set data pelatihan yang buruk dapat menghasilkan kesalahan prediksi dalam model yang dilakukan deployment. Inilah alasan utama para ilmuwan data menghabiskan sebagian besar waktu mereka untuk menyiapkan data untuk ML.

**„** 80 percent of a data scientist's valuable time is spent simply finding, cleansing, and organizing data, leaving only 20 percent to actually perform analysis...

IBM Data Analytics



# TEKNIK CLEANSING DATA

- Handling Null Values
- Handling Outliers
- Checking Data Types
- Dropping Irrelevant Columns

# HANDLING NULL VALUES

Null values atau missing values adalah nilai yang hilang atau tidak tersedia dalam dataset, biasa disebut NaN. Hal ini bisa terjadi karena berbagai alasan, seperti data yang tidak diisi saat pengumpulan atau kesalahan input.

Jika null values tidak ditangani, hal ini dapat menyebabkan bias dalam analisis, mengganggu performa model machine learning, atau bahkan membuat algoritma gagal bekerja.

## Metode:

- **Data Deletion:** Menghapus data
- **Data Imputation:** Mengisi nilai null dengan mean/median/modus/custom imputation

# Drop NaN value

The diagram illustrates the process of dropping NaN values from a DataFrame. On the left, a DataFrame with four rows and three columns (col\_a, col\_b, col\_c) is shown. The first row has col\_a=1.0, col\_b=5.0, and col\_c=9. The second row has col\_a=2.0, col\_b=NaN, and col\_c=10. The third row has col\_a=NaN, col\_b=NaN, and col\_c=11. The fourth row has col\_a=4.0, col\_b=8.0, and col\_c=12. An arrow points from the original DataFrame to a modified one on the right. The modified DataFrame has only two rows. The first row (index 0) contains col\_a=1.0, col\_b=5.0, and col\_c=9. The second row (index 3) contains col\_a=4.0, col\_b=8.0, and col\_c=12. The rows with NaN values have been removed.

	col_a	col_b	col_c
0	1.0	5.0	9
1	2.0	NaN	10
2	NaN	NaN	11
3	4.0	8.0	12

The diagram illustrates the process of filling NaN values in a DataFrame. On the left, a DataFrame with eight rows and four columns (date, fruit, price) is shown. The first four rows represent apples: date 2021-01-01 (price 0.8), date 2021-01-02 (price NaN), date 2021-01-03 (price NaN), and date 2021-01-04 (price 1.2). The next four rows represent mangoes: date 2021-01-01 (price NaN), date 2021-01-02 (price 3.1), date 2021-01-03 (price NaN), and date 2021-01-04 (price 2.8). Orange arrows labeled "fill" point from the NaN cells in the apple rows to the corresponding cells in the mango rows, indicating that the missing values are being replaced by the last available value from the previous group. An orange arrow also points from the last NaN cell in the mango group to the final filled DataFrame on the right.

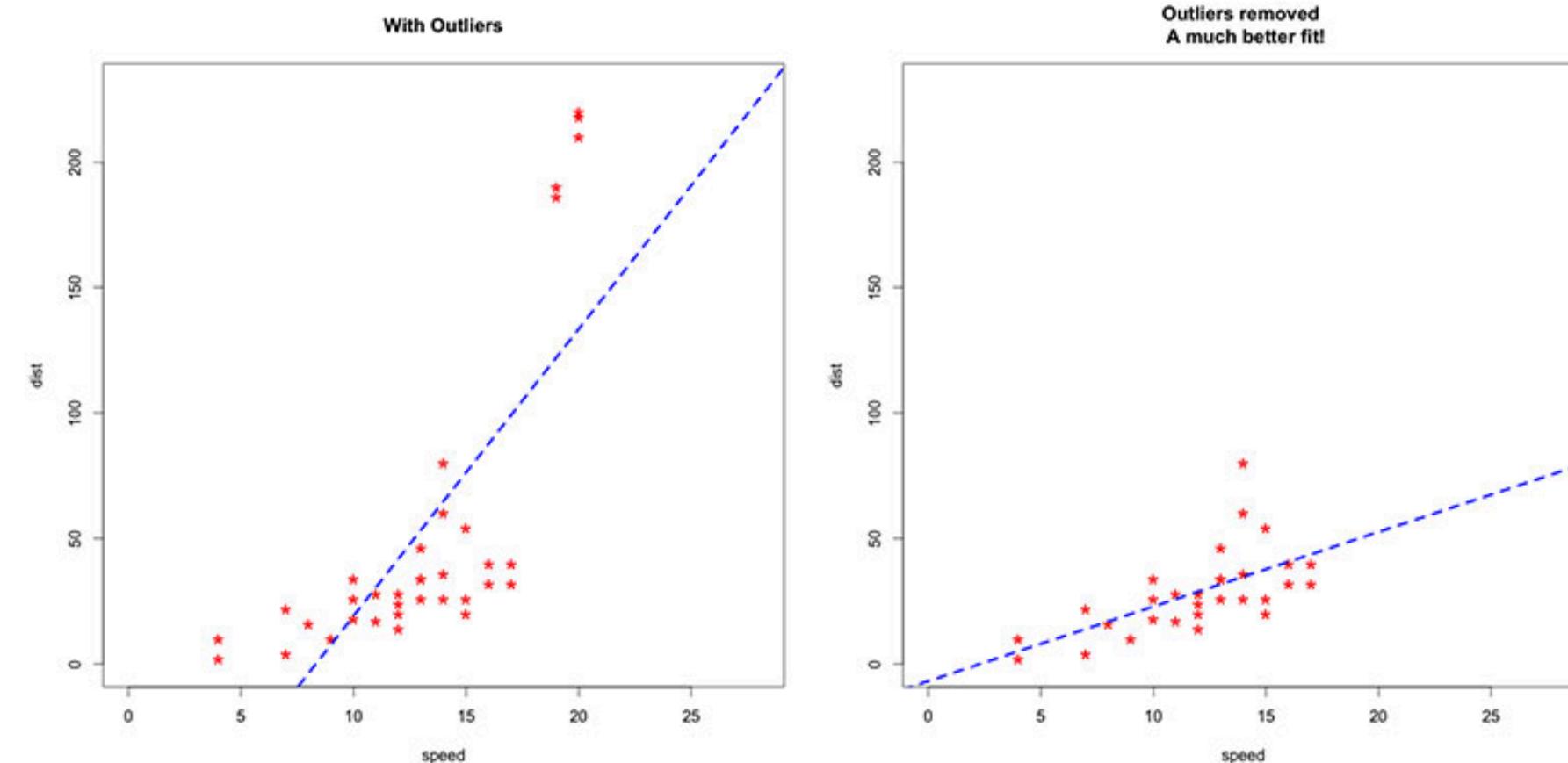
	date	fruit	price
0	2021-01-01	apple	0.8
1	2021-01-02	apple	NaN
2	2021-01-03	apple	NaN
3	2021-01-04	apple	1.2
4	2021-01-01	mango	NaN
5	2021-01-02	mango	3.1
6	2021-01-03	mango	NaN
7	2021-01-04	mango	2.8

# Fill NaN value

- Mean:** Distribusi data normal/mendekati, tidak ada outlier signifikan
- Median:** Distribusi miring (skewed) terdapat outlier signifikan
- Modus:** Data kategorikal, ada kategori dominan
- Costum:** Nilai relevan dengan data

# HANDLING OUTLIERS

Outlier adalah nilai atau data point yang sangat berbeda atau jauh dari kebanyakan data lainnya dalam sebuah dataset. Outlier bisa disebabkan oleh kesalahan pengukuran, kesalahan input data, atau memang merupakan hasil yang sebenarnya. Penting untuk menangani outlier dengan benar karena outlier dapat mempengaruhi hasil analisis dan model machine learning secara signifikan.



# Metode

- **Remove:** Menghapus Outlier
  - Ketika disebabkan salah pengukuran atau input
  - Tidak merepresentasikan data sebenarnya atau tidak relevan
  - Dataset cukup besar sehingga tidak terlalu berpengaruh
- **Transform:** Mengubah skala data
  - Ketika outlier merupakan bagian dari distribusi data
  - Outlier berpengaruh terhadap hasil atau analisis
  - Distribusi data miring (skewed): Log Transformation
  - Menormalisasikan distribusi: Box-Cox
- **Cap/Floor:** Pembatasan Nilai
  - Ketika ingin membatasi pengaruh outlier tanpa menghapus data.
  - Cocok untuk data yang memiliki rentang nilai yang jelas, dan outlier berada di luar rentang tersebut.
  - Ketika outlier masih relevan tetapi perlu dibatasi dampaknya.
- **Imputation:** Mengganti nilai
  - Ketika outlier tidak bisa dihapus dan tetap ingin mempertahankan data tanpa mengubah terlalu banyak.
  - Saat menghindari distorsi yang disebabkan oleh outlier dengan menggantinya dengan nilai yang lebih wajar.

# CHECKING DATA TYPE

Checking Data Type atau memeriksa tipe data dalam pengolahan data mengacu pada proses untuk memastikan bahwa tipe data setiap kolom dalam DataFrame atau dataset sesuai dengan yang diharapkan. Memeriksa tipe data penting karena tipe data yang salah dapat menganggu analisis data dan pemodelan yang dilakukan.

	Nama	Usia	Tinggi
0	Alice	25	160.5
1	Bob	30	175.0
2	Charlie	35	168.3

```
Nama          object  
Usia         int64  
Tinggi      float64  
dtype: object
```

# DROPPING IRRELEVANT COLUMNS

Tidak semua kolom dalam dataset memiliki nilai atau kontribusi yang signifikan untuk analisis atau model machine learning. Oleh karena itu, sangat penting untuk mengidentifikasi dan menghapus kolom-kolom yang tidak relevan.

## Mengapa Menghapus Kolom yang Tidak Diperlukan?

- Meningkatkan Kinerja Model
- Mengurangi Dimensi Data

## Jenis Kolom yang Mungkin dihapus

- Kolom dengan nilai konstan
- Kolom dengan data yang tidak relevan
- Kolom dengan banyak null

## Mana yang Tidak Relevan?

ID	Name	Age	Gender	Status
1	Alice	25	F	Active
2	Bob	30	M	Active
3	Charlie	35	M	Active
4	David	40	M	Active
5	Eve	45	F	Active

# **DATA PREPROCESSING**

Data preprocessing adalah teknik yang digunakan untuk mengubah data mentah dalam format yang berguna dan efisien. Tahap ini diperlukan karena data mentah seringkali tidak lengkap dan memiliki format yang tidak konsisten.

## **Teknik Preprocessing**

- Scalling
- Sampling
- Feature Engineering
- Encoding

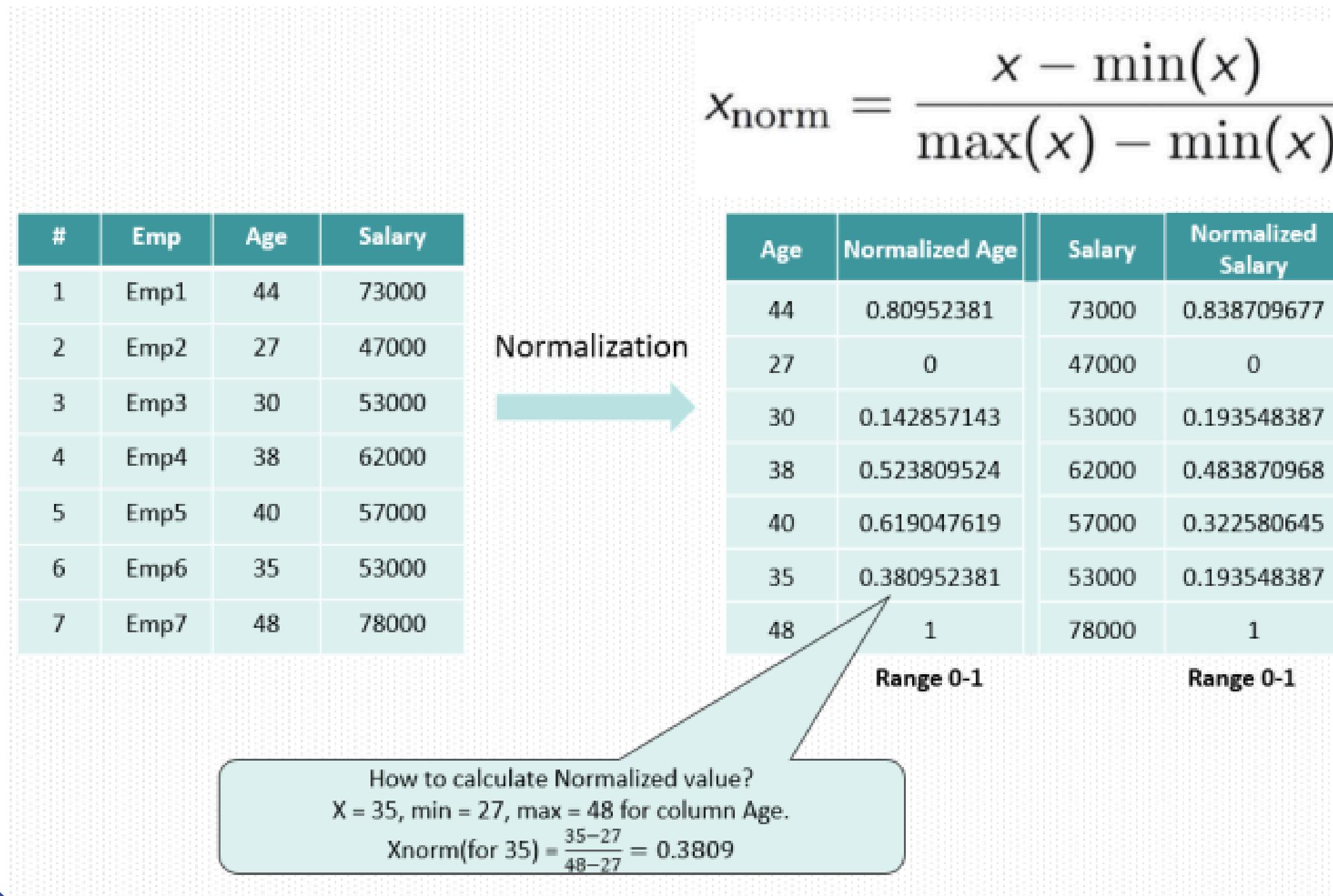
# SCALING

Scaling menyesuaikan rentang fitur agar kontribusinya seimbang dalam analisis atau model. Ini penting untuk algoritma yang bergantung pada pengukuran jarak atau menganggap data terdistribusi normal.

## Teknik:

- **Min-Max Scaling:** Mengubah fitur menjadi rentang antara 0 sampai 1 atau -1 sampai 1 (bila ada data negatif)
- **Standardisasi:** Mengubah mean menjadi 0 dan deviasi standar menjadi 1
- **Robust Scaling:** Menghapus median dan mengukur data di rentang antara kuartil-1 dan kuartil ke-3

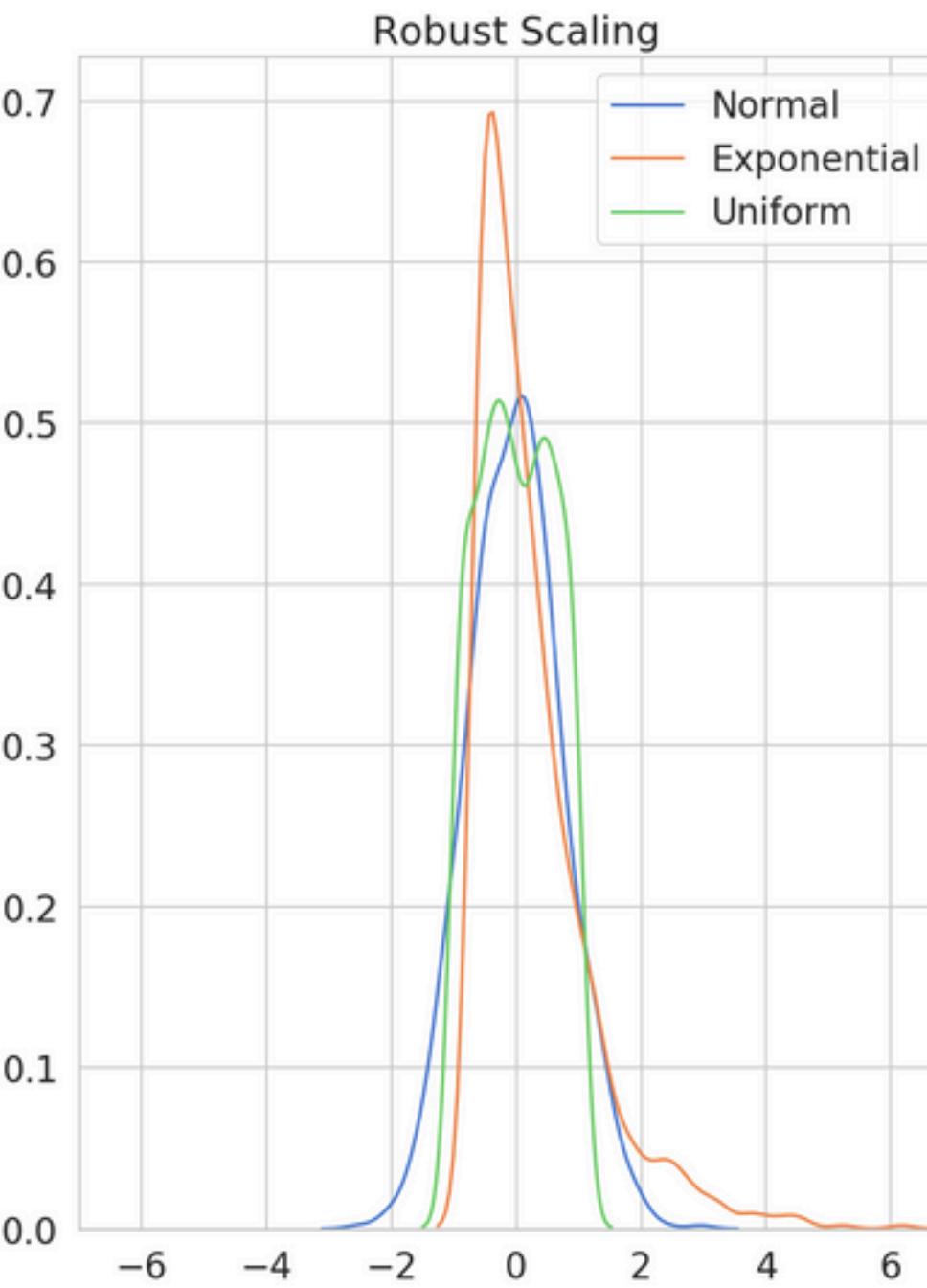
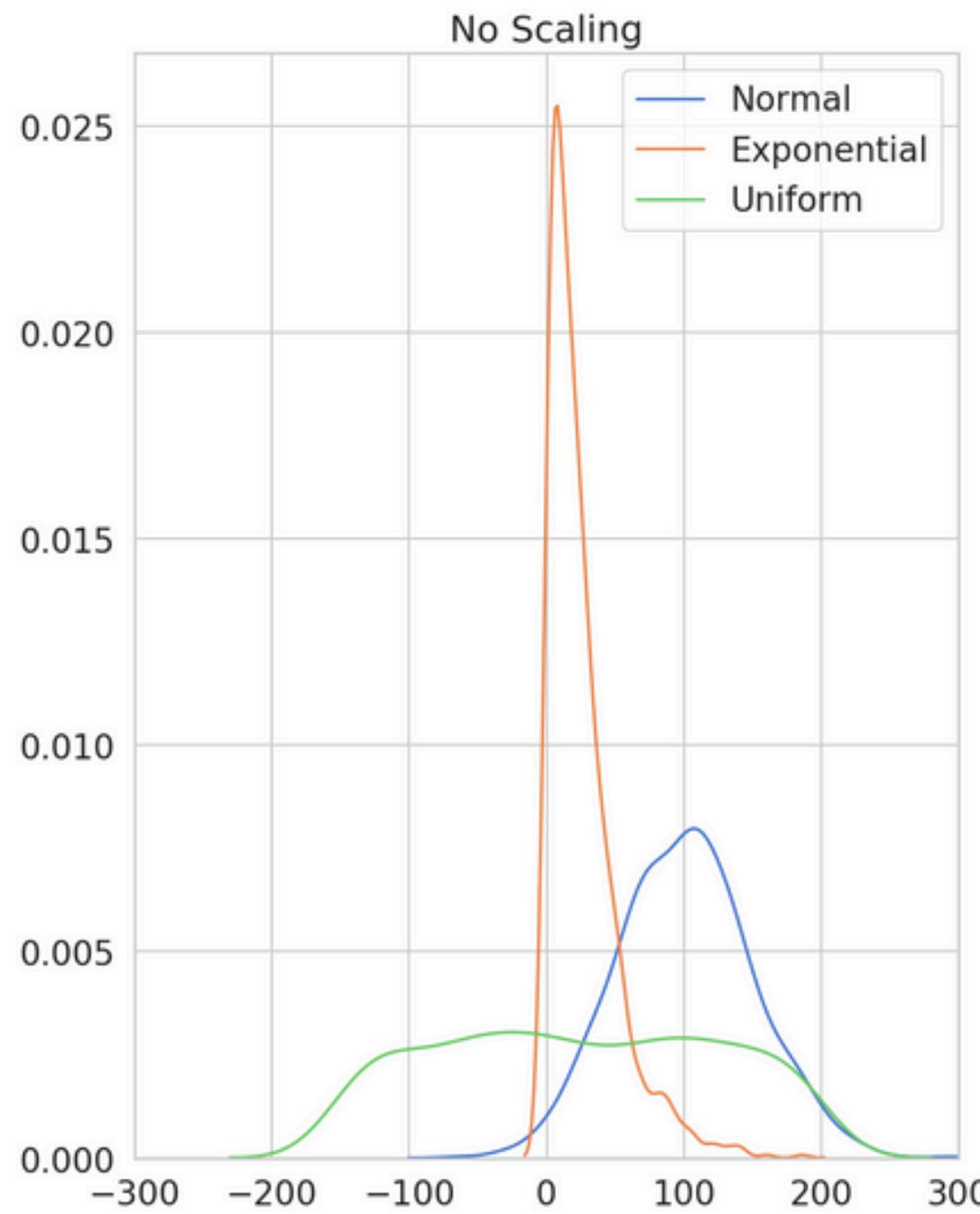
# Min-Max Scaling



## Cocok untuk data:

- Algoritma model distance-based (KNN,SVM)
- Distribusi data tidak normal

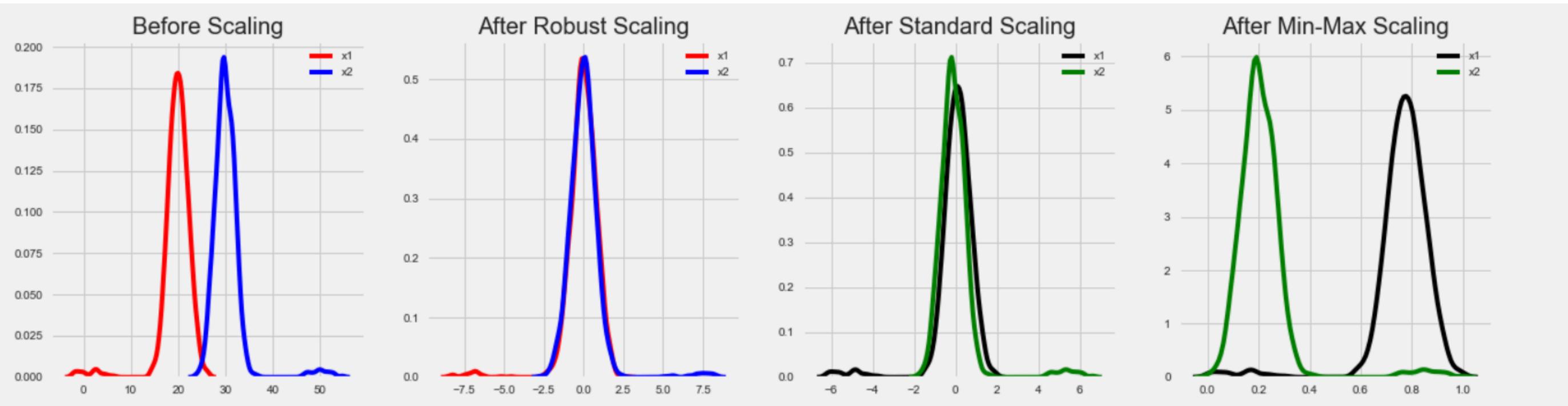
# Robust Scaling



$$X_{\text{scale}} = \frac{x_i - x_{\text{med}}}{x_{75} - x_{25}}$$

## Cocok untuk data:

- Terdapat outlier yang signifikan dan dapat mempengaruhi hasil scaling
- Algoritma yang terpengaruh oleh outliers (SVM, KNN)



1	Original	Standardization	Max-Min Scaler	Rubost Scaler
2	6.9314183	-0.2244971	0.0000003	0.8283487
3	2.6674115	-0.2244979	0.0000001	0.0690181
4	7.7248183	-0.2244970	0.0000003	0.9696367
5	5.7388433	-0.2244973	0.0000002	0.6159760
6	0.8965615	-0.2244982	0.0000000	-0.2463333
7	4.5147618	-0.2244975	0.0000002	0.3979926
8	2.9934144	-0.2244978	0.0000001	0.1270724
9	4.8708377	-0.2244975	0.0000002	0.4614023
10	4.2797819	-0.2244976	0.0000002	0.3561476
11	1.0085616	-0.2244982	0.0000000	-0.2263885
12	5.5166580	-0.2244974	0.0000002	0.5764094
13	1.1171326	-0.2244981	0.0000000	-0.2070542
14	0.4069897	-0.2244983	0.0000000	-0.3335159
15	5.0536949	-0.2244975	0.0000002	0.4939654
16	8.4068370	-0.2244969	0.0000003	1.0910900
17	8.9588050	-0.2244968	0.0000003	1.1893840
18	0.9543401	-0.2244982	0.0000000	-0.2360442
19	94750.5292279	-0.2079018	0.0037104	16872.6857158
20	2051.2433203	-0.2241390	0.0000803	364.8776314
21	25536631.9371928	4.2485000	1.0000000	4547540.7645023

# SAMPLING

Teknik sampling digunakan untuk menangani dataset yang tidak seimbang atau mengurangi ukuran dataset besar agar lebih mudah dikelola.

## Teknik:

- **Oversampling:** Meningkatkan jumlah instance pada kelas minoritas.
- **Undersampling:** Mengurangi jumlah instance pada kelas mayoritas.
- **Random Sampling:** Pengambilan sampling secara acak dan memiliki peluang yang sama.

## Simple Random Sample



# Random Sampling

- Ketika dataset terlalu besar dan ingin mengambil sampel secara acak
- Digunakan dalam tahap awal analisis data untuk mendapatkan gambaran umum dari dataset secara cepat.

## Under dan Over Sampling

Undersampling



Oversampling



- Ketika jumlah dataset tidak seimbang
- Digunakan ketika model machine learning yang digunakan sensitif terhadap ketidakseimbangan

# FEATURE ENGINEERING

Proses di mana data mentah diubah atau diperkaya menjadi fitur-fitur yang lebih representatif dan relevan untuk model machine learning. Tujuan utama dari feature engineering adalah untuk menciptakan fitur yang dapat meningkatkan kinerja model dengan memberikan informasi yang lebih berguna untuk membuat prediksi.

## Teknik:

- **Feature Selection:** Menghilangkan fitur yang tidak relevan dan memilih fitur yang relevan
- **Feature Creation:** Membuat fitur baru dari kombinasi fitur lain
- **Feature Splitting:** Memecah suatu fitur menjadi fitur baru

	34	"	Age	Income	Education	City	Gender	Productivity\
	35	"0	25	50000	High School	New York	Male	5\
	36	"1	30	60000	Bachelor	San Francisco	Female	4\
	37	"2	35	75000	Master	Chicago	Male	3\
	38	"3	40	90000	Ph.D.	Los Angeles	Female	2\
	39	"4	45	80000	Bachelor	Miami	Male	4\

## Feature Creation dan Splitting

Out[8]:

	City_Miami	City_New York	City_San Francisco	Gender_Female	Gender_Male	Income per Age
	False	True	False	False	True	1.039655
	False	False	True	True	False	1.089162
	False	False	False	False	True	inf
	False	False	False	True	False	1.881280
	True	False	False	False	True	0.445566

## Feature Selection

```
Selected Features: Index(['Income', 'Gender_Male', 'Income per Age'], dtype='object')
```

# ENCODING

proses mengubah data kategorikal menjadi format numerik yang dapat dipahami oleh model machine learning karena banyak algoritma machine learning tidak dapat bekerja langsung dengan data kategorikal (misalnya, nama kota, jenis kelamin, warna, dll.). Encoding diperlukan untuk mengubah kategori tersebut menjadi angka.

## Teknik:

- **Label Encoding:** Mengubah nilai suatu kolom menjadi bentuk numerik unik tanpa memperhatikan urutan
- **One-Hot Encoding:** Membuat kolom baru berdasarkan value dari suatu kolom dan mengisinya dengan bentuk biner (0 dan 1)
- **Ordinal Encoding:** Mengubah nilai suatu kolom menjadi bentuk numerik unik dengan memperhatikan urutan

# Label Encoding

State (Nominal Scale)
Maharashtra
Tamil Nadu
Delhi
Karnataka
Gujarat
Uttar Pradesh

State (Label Encoding)
3
4
0
2
1
5

- Digunakan ketika kategori tidak memiliki urutan yang jelas dan jumlah yang cukup banyak.

# Ordinal Encoding

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

- Digunakan untuk fitur kategorikal yang memiliki hierarki atau urutan jelas.

# One-Hot Encoding

id	color
1	red
2	blue
3	green
4	blue

One-Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

- Ideal saat kategori tidak memiliki urutan khusus dan jumlah kategori tidak terlalu banyak.

**WEEK 4**

# **MACHINE LEARNING MODELING, METRICS, AND OPTIMIZATION**

**MENTOR : RAP**



“Predicting the future isn’t magic,  
it’s artificial intelligence.”

- Dave Waters

# Traditional Programming



```
if v <= 5 :  
    return Walk
```

**v = Velocity (m/s)**



```
if v <= 20 :  
    return Run  
  
elif v <= 5 :  
    return Walk
```



```
if v <= 40 :  
    return Cycling  
  
elif v <= 20 :  
    return Run  
  
elif v <= 5 :  
    return Walk
```



???

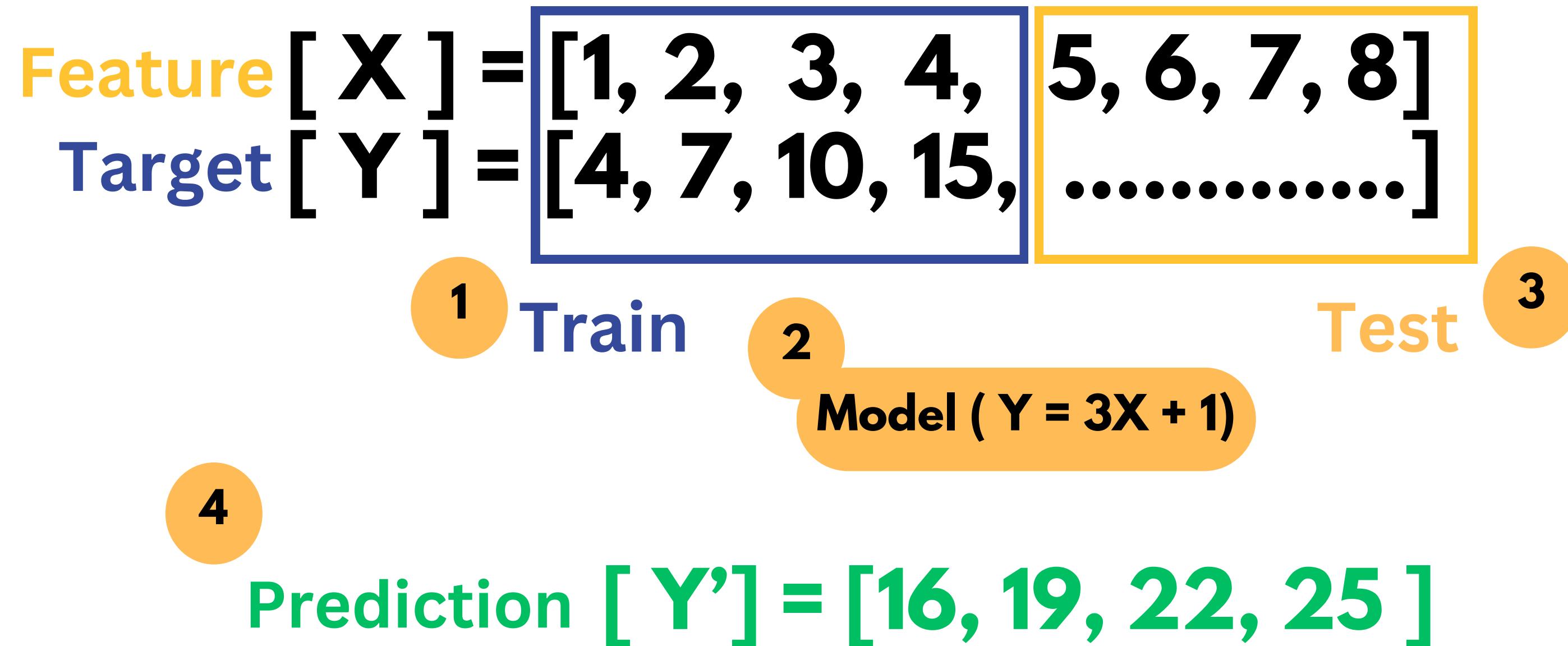
Dalam pemrograman tradisional kita **menentukan sebuah rules** untuk mendapat sebuah keputusan

# WHAT IS MACHINE LEARNING ?

Machine Learning adalah sistem pemrograman yang terinspirasi dari cara belajar manusia, yang memungkinkan komputer menentukan keputusan atau **membuat prediksi berdasarkan data input dan output.**

$$\begin{aligned}[X] &= [1, 2, 3, 4, 5, 6, 7, 8] \\ [Y] &= [4, 7, 10, 15, \dots]\end{aligned}$$

# How MACHINE LEARNING works ?



1

# Splitting Data

[ X ] = [ 1, 2, 3, 4, 5, 6, 7, 8 ]

[ Y ] = [ 4, 7, 10, 15, 16, 19, 22, 25 ]

Kunci Jawaban

Train Set

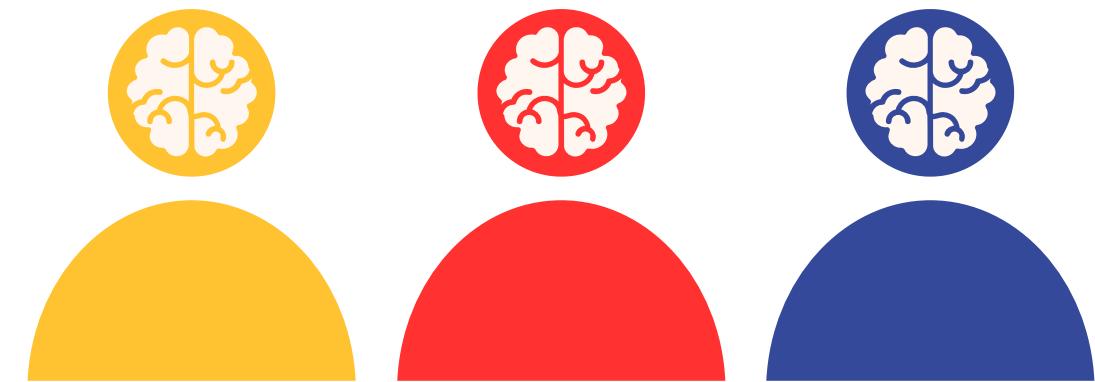
Test Set

Usahakan splitting data dengan kondisi Train lebih banyak daripada Test supaya model lebih banyak belajar

2

## Model Selection

Model selection ibarat kamu memilih orang yang hendak kamu suruh untuk mencari pola atau rumus dari data yang kamu punya



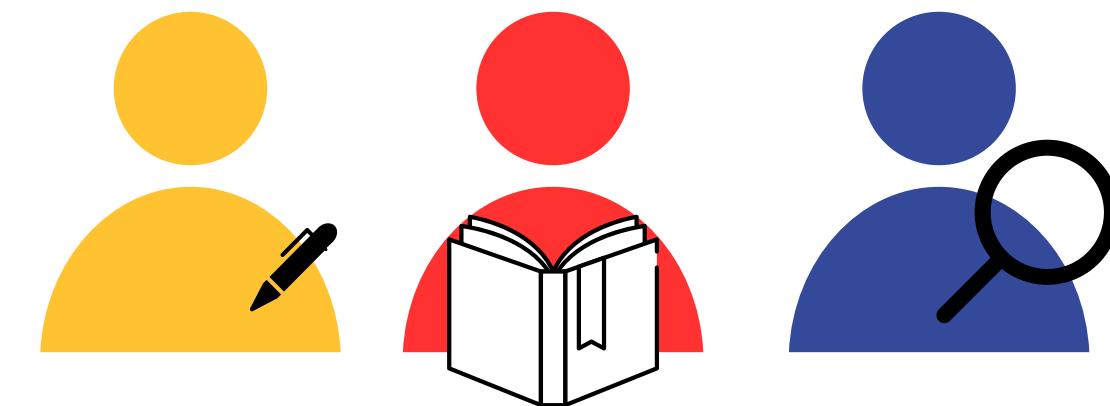
Setiap orang memiliki cara berpikir yang berbeda, jadi usahakan pilih yang paling tepat dan pilih lebih dari satu supaya bisa dibandingkan

3

## Training Model

### Train Set

$$\begin{aligned} [X] &= [1, 2, 3, 4, 5, ] \\ [Y] &= [4, 7, 10, 15, 16, ] \end{aligned}$$



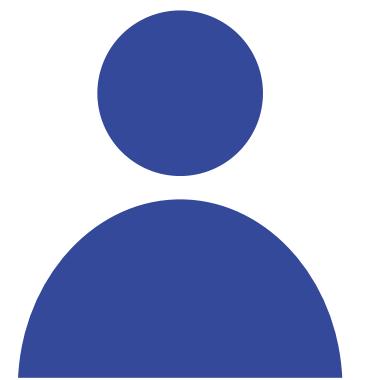
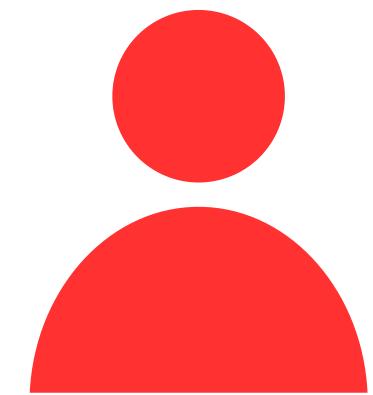
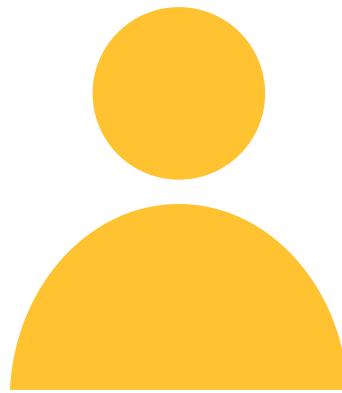
Setiap orang mempelajari pola dan rumus dari data train yang diberikan dan menetapkan rumus tersebut untuk mengerjakan test set

4

## Testing Model

$$[X] = [6, 7, 8]$$

$$[Y'] = [20, 21, 23] \quad [Y'] = [19, 22, 25] \quad [Y'] = [18, 22, 24]$$



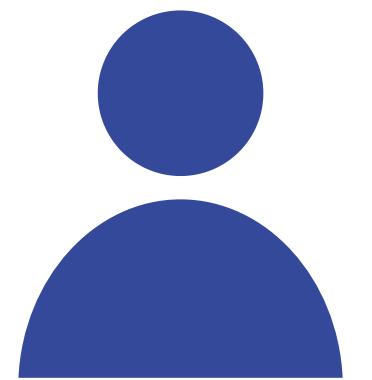
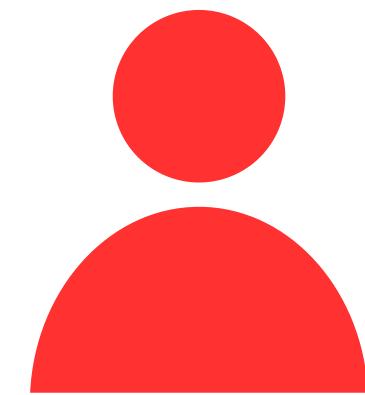
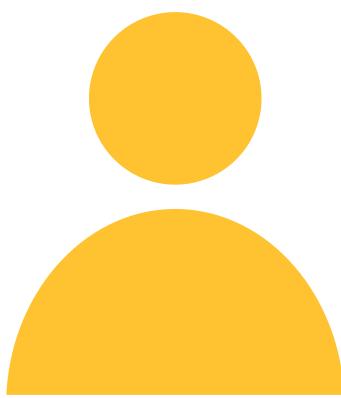
Setiap orang menjawab atau memprediksi nilai Y yang seharusnya muncul dengan pola dan rumus yang telah mereka tetapkan sebelumnya

5

## Evaluation

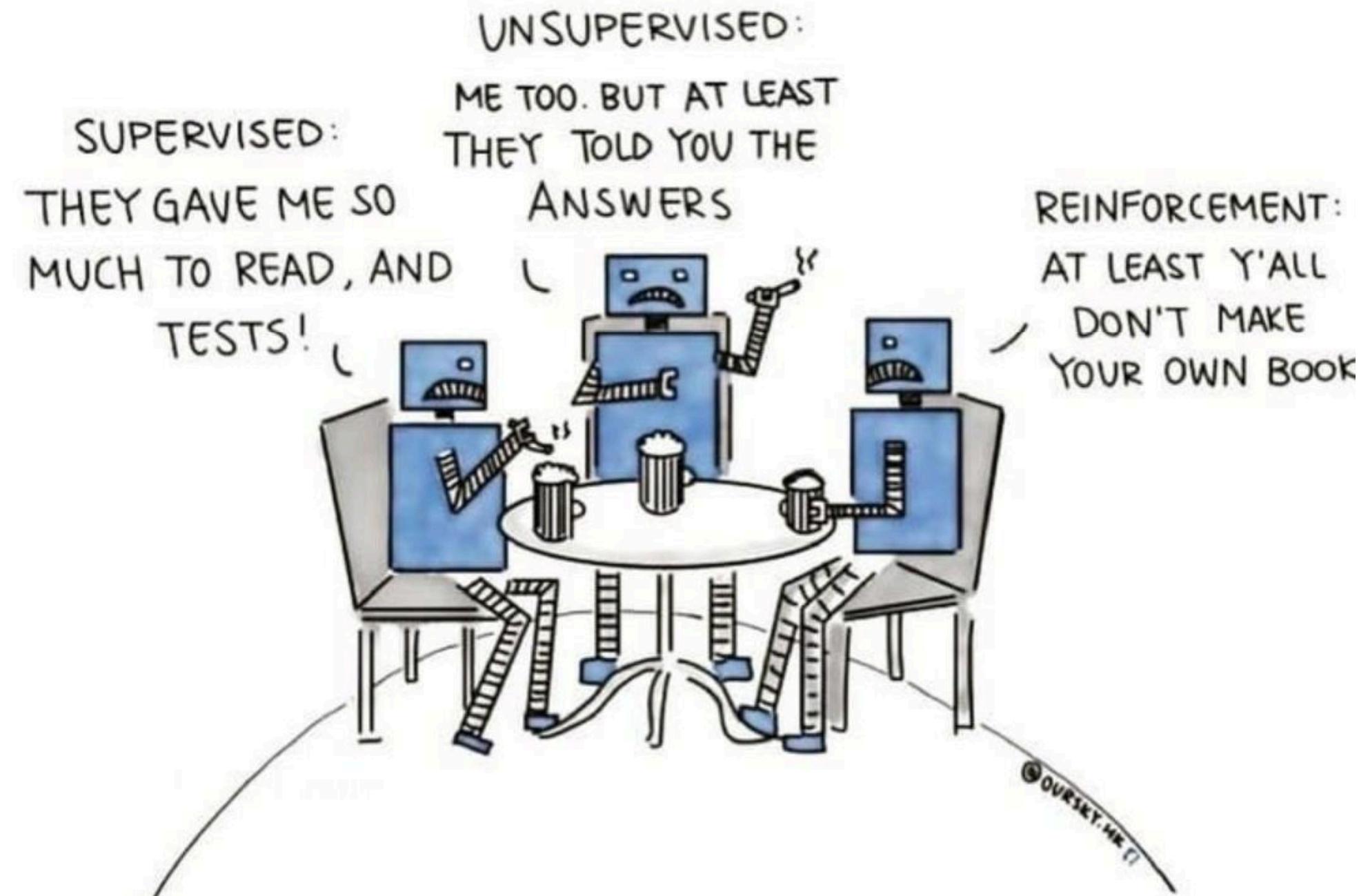
$$[Y] = [19, 22, 25]$$

$$[Y'] = [20, 21, 23] \quad [Y'] = [19, 22, 25] \quad [Y'] = [18, 22, 24]$$



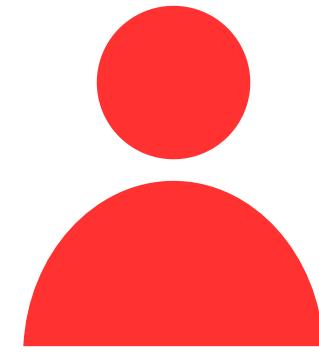
Kamu membandingkan jawaban tiap orang dengan kunci jawaban yang kamu miliki dan melakukan penilaian seberapa baik mereka menjawab

# Three main types of Machine Learning Algorithms



## Training Model

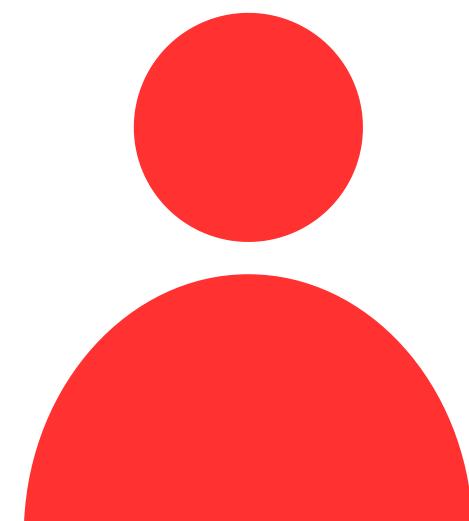
[ X ] = [cat, dog, lizard, snake, dolphin, shark]



model mempelajari value data yang diberikan dan mempelajari hubungan tiap nilai dengan nilai-nilai lainnya

## Clustering

[ X ] = [cat, dog, lizard, snake, dolphin, shark]



[ A ] = [cat, dog]  
[ B ] = [lizard, snake]  
[ C ] = [dolphin, shark]

model membagi tiap nilai menjadi beberapa cluster dengan mempertimbangkan hubungan antar nilai

# EVALUATION METRICS

# METRICS ?

**Metric** dalam konteks machine learning adalah ukuran yang digunakan untuk mengevaluasi kinerja model. Metrik membantu kita memahami seberapa baik model kita dalam membuat prediksi dan seberapa akurat atau efektif hasil yang diberikan oleh model tersebut.

## Mengapa Metric Penting?

- **Evaluasi Model:** Metric memberi kita cara objektif untuk menilai kualitas model. Tanpa metric, kita tidak tahu seberapa baik model kita dibandingkan dengan model lain atau dengan hasil yang diharapkan.
- **Perbandingan Model:** Dengan metric, kita bisa membandingkan berbagai model atau algoritma untuk melihat mana yang memberikan hasil terbaik.
- **Optimasi Model:** Metric membantu kita memahami area mana dari model yang perlu diperbaiki.

## SUPERVISED

### Regression   Classification

- MAE
- MSE
- RMSE
- $R^2$

### Confusion Matrix

## UNSUPERVISED

- Silhouette Score
- Davies-Bouldin Index
- Elbow Method

# **REGRESSION METRICS**

Metrics yang dipakai untuk mengukur seberapa jauh nilai prediksi dengan data aktual

# MAE

## Mean Absolute Error

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

### Level : Kurang Terasa

MAE menghitung rata-rata nilai absolut dari kesalahan. Ini memberikan ukuran kesalahan yang lebih stabil karena tidak mengkuadratkan kesalahan, sehingga setiap kesalahan besar memiliki dampak yang sama pada MAE seperti kesalahan kecil. MAE lebih "rata" dan **tidak terlalu dipengaruhi oleh outlier** dibandingkan dengan MSE dan RMSE.

# RMSE

## Root Mean Squared Error

$$\text{RMSE} = \sqrt{\text{MSE}}$$

### Level : Menengah

RMSE adalah akar kuadrat dari MSE, sehingga nilainya kembali dalam satuan yang sama dengan data asli. RMSE memberikan gambaran yang lebih jelas tentang seberapa besar kesalahan rata-rata, tetapi masih mempertimbangkan dampak kesalahan besar karena MSE juga **sensitif terhadap outlier**.

# MSE

## Mean Squared Error

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

### Level : Paling Terasa

MSE mengkuadratkan kesalahan, sehingga kesalahan yang lebih besar memberikan kontribusi yang jauh lebih besar pada nilai MSE. Dengan kata lain, kesalahan besar dihukum lebih berat, dan ini membuat MSE **sangat sensitif terhadap outlier** (data yang sangat berbeda dari yang lain).

# R<sup>2</sup> Score

## Residual Sum of Squares (SSE):

$$SST = \sum (y_i - \bar{y})^2$$

di mana  $y_i$  adalah nilai aktual dan  $\bar{y}$  adalah rata-rata nilai aktual.

## Total Sum of Squares (SST):

$$SSE = \sum (y_i - \hat{y}_i)^2$$

di mana  $\hat{y}_i$  adalah nilai yang diprediksi oleh model.

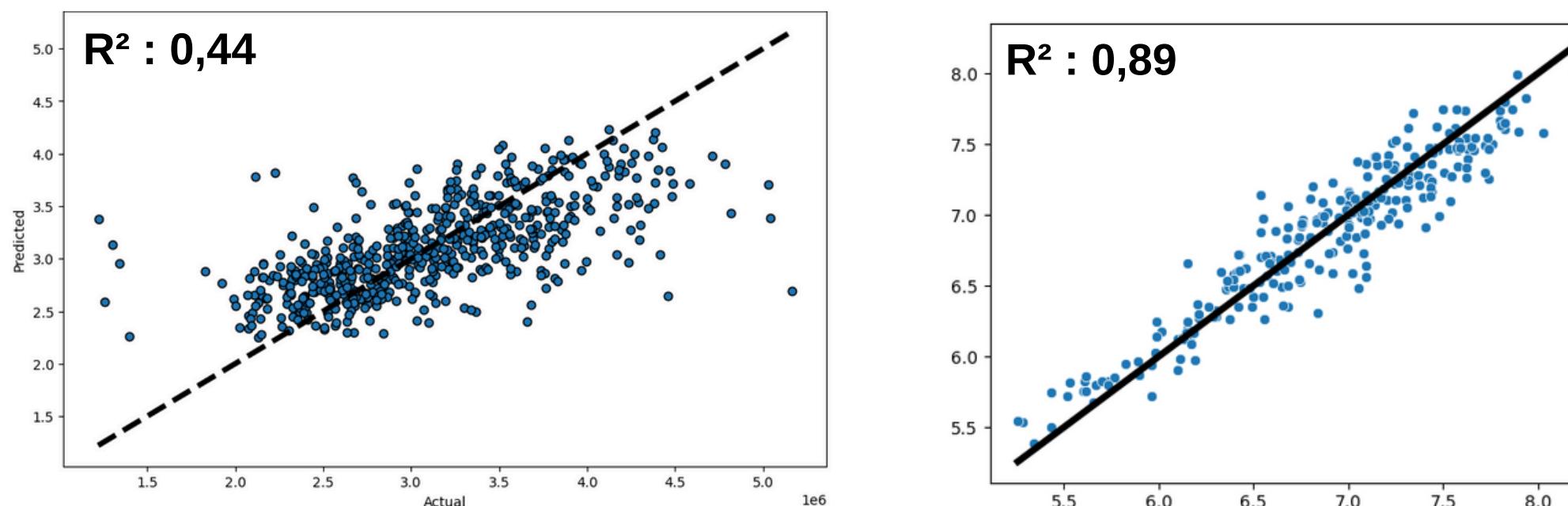
## R-Squared Score

$$R^2 = 1 - \frac{SSE}{SST}$$

R-squared ( $R^2$ ), juga dikenal sebagai koefisien determinasi, adalah ukuran statistik yang digunakan untuk mengevaluasi seberapa baik model regresi menjelaskan variasi dalam data. Ini memberikan indikasi seberapa baik model fit dengan data yang sebenarnya.

- **Tidak Menunjukkan Kualitas Model Secara Keseluruhan:** R-squared tidak memberikan informasi tentang seberapa baik model fit dengan data. Model dengan R-squared tinggi bisa jadi overfitting, terutama jika ada banyak variabel.
- **Sensitif Terhadap Outlier:** Outlier dapat mempengaruhi R-squared, membuatnya tampak lebih baik atau lebih buruk dari yang sebenarnya.
- **Tidak Berlaku untuk Semua Jenis Model:** R-squared paling sering digunakan dalam regresi linear. Untuk model yang lebih kompleks atau non-linear, seperti regresi logistik, R-squared mungkin tidak memberikan informasi yang berarti.

R-Squared biasanya dibuktikan menggunakan Scatter Plot dengan membandingkan data aktual dengan prediksi



# **CLASSIFICATION METRICS**

# What is Confusion Matrix ?



Matrix Kebingungan???

# Confusion Matrix

		Truth	
		1	0
Prediction	1	TP	FP
	0	FN	TN

**Analogi** : Bayangkan kamu lagi suka sama seseorang, terus kamu berpikir apakah dia juga suka kamu atau nggak? terus bagaimana kenyataannya? dari semua pemikiran dan kenyataan itu kita pecah jadi 4 kondisi

## Positive Thinking

- **True Positive** : Kamu berpikir dia suka kamu dan ternyata kenyatannya dia memang suka kamu
- **False Positive** : Kamu berpikir dia suka kamu dan ternyata kenyatannya dia gak suka kamu

## Negative Thinking

- **False Negative** : Kamu berpikir dia gak suka kamu tetapi ternyata kenyatannya dia suka kamu
- **True Negative** : Kamu berpikir dia gak suka kamu dan memang kenyatannya dia gak suka kamu

## Accuracy

		Truth	
		Sakit	Sehat
Pred	Sakit	10	20
	Sehat	30	80

$$\frac{10 + 80}{10 + 80 + 20 + 30} = 0,64$$

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

Accuracy: Proporsi prediksi yang benar dibandingkan dengan total prediksi.

**Accuracy bukan segalanya dalam menilai sebuah model klasifikasi**

## Recall (Pengukur sensitivitas)

		Truth	
		Sakit	Sehat
Pred	Sakit	10	20
	Sehat	30	80

$$\frac{10}{10 + 30} = 0,25$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall : Proporsi nilai positif yang benar-benar terdeteksi oleh model

**Pada beberapa kasus, model yang memiliki nilai recall yang kecil dapat membuat keputusan atau predikksi yang diambil berbahaya**

## Precision

		Truth	
		Bersalah	Tidak Bersalah
Pred	Bersalah	10	40
	Tidak Bersalah	30	80

$$\frac{10}{10 + 40} = 0,20$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision : proporsi prediksi positif yang benar dibandingkan dengan total prediksi positif.

**Pada beberapa kasus, model yang memiliki nilai Precision yang kecil dapat membuat keputusan atau prediksi yang diambil berbahaya**

## F1 Score

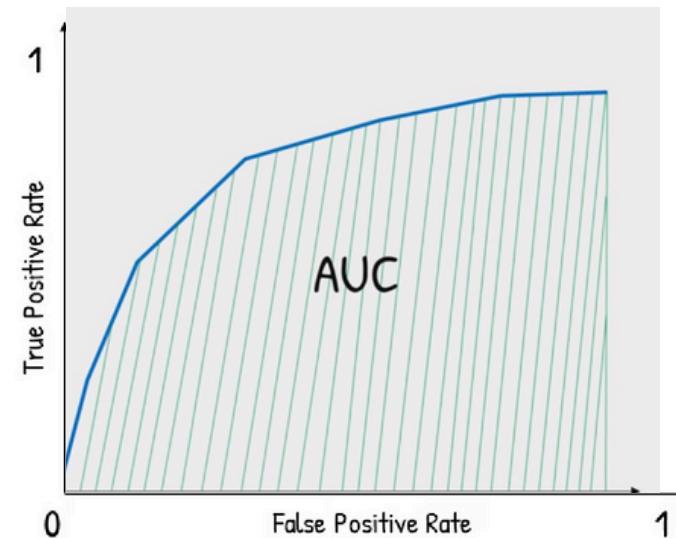
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 Score: Harmonik rata-rata antara precision dan recall

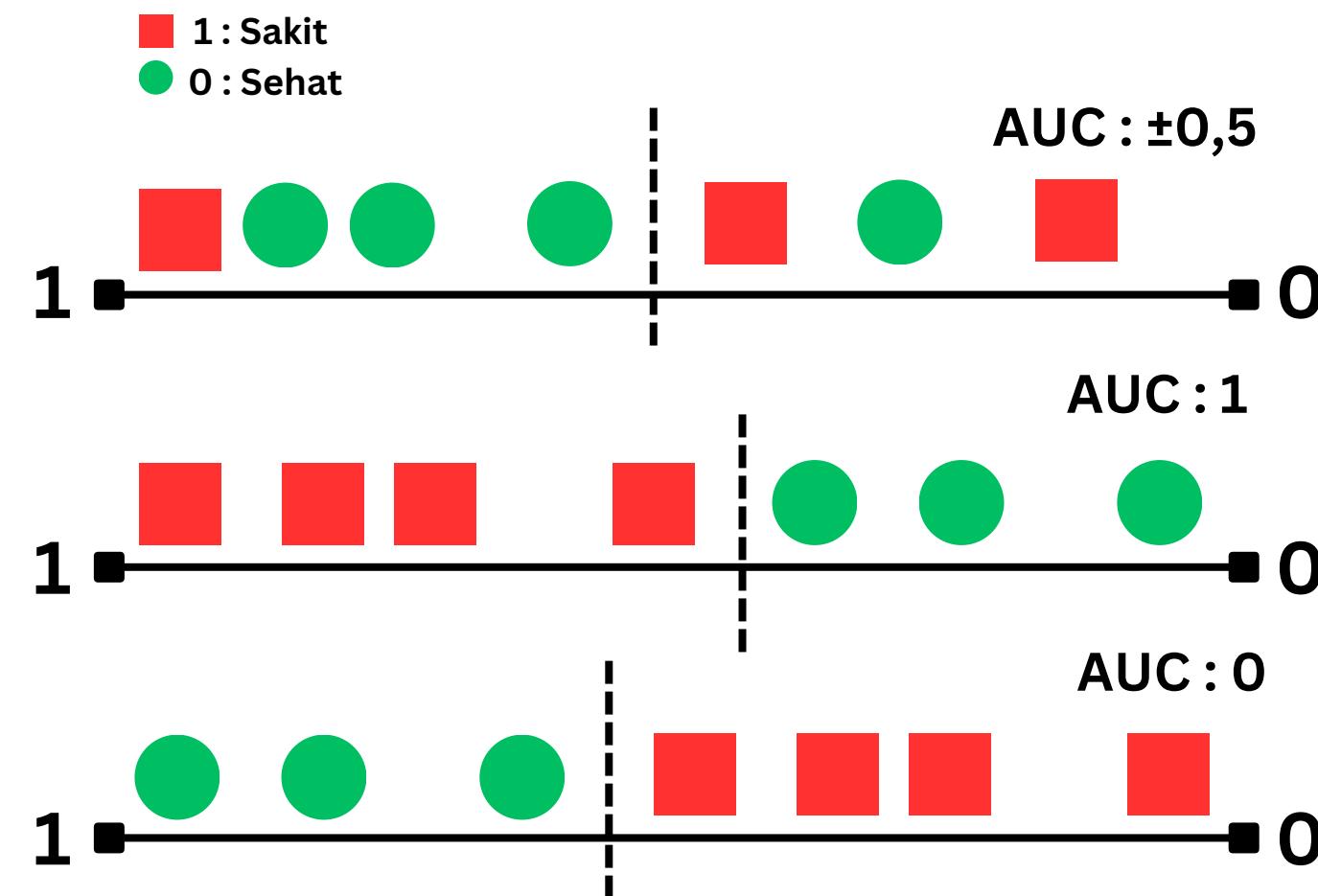
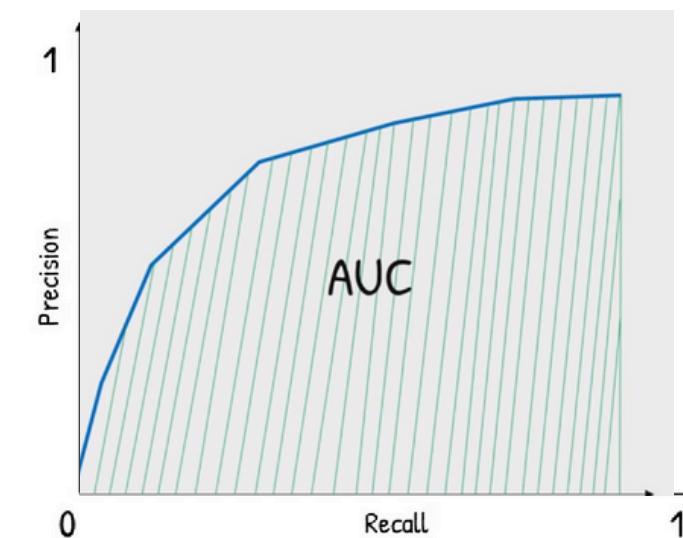
Untuk mengetahui seberapa seimbang antara Precision dan Recall pada sebuah model

# AUC : Area Under Curve

**ROC Curve**  
Receiver Operating Characteristic



**PR Curve**  
Precision Recall



## Kapan ROC Curve Lebih Dipertimbangkan:

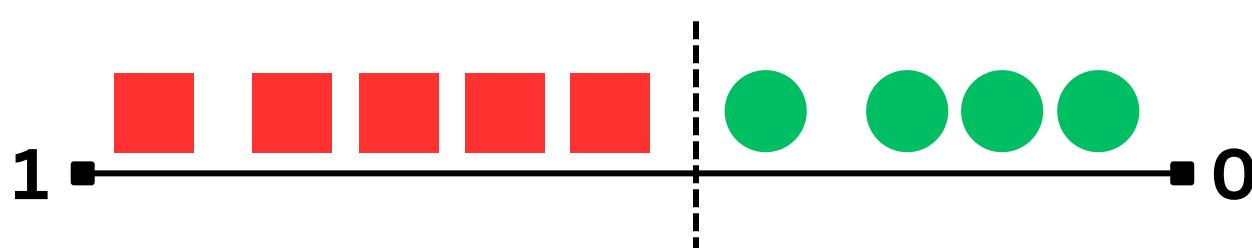
- Ketika Dataset Seimbang atau Mendekati Seimbang:**

ROC curve umumnya lebih informatif ketika dataset memiliki distribusi kelas yang seimbang atau mendekati seimbang antara kelas positif dan negatif. Hal ini karena ROC curve mempertimbangkan baik true positives (TP) maupun false positives (FP) secara proporsional.

- Ketika False Positives dan False Negatives Sama Pentingnya:**

Jika kesalahan dalam mengklasifikasikan kelas positif sebagai negatif (false negatives) sama pentingnya dengan kesalahan mengklasifikasikan kelas negatif sebagai positif (false positives), maka ROC curve memberikan gambaran yang lebih komprehensif tentang performa model di semua threshold.

Keyword : **Class Negative** diperhatikan



## Kapan PR Curve Lebih Dipertimbangkan:

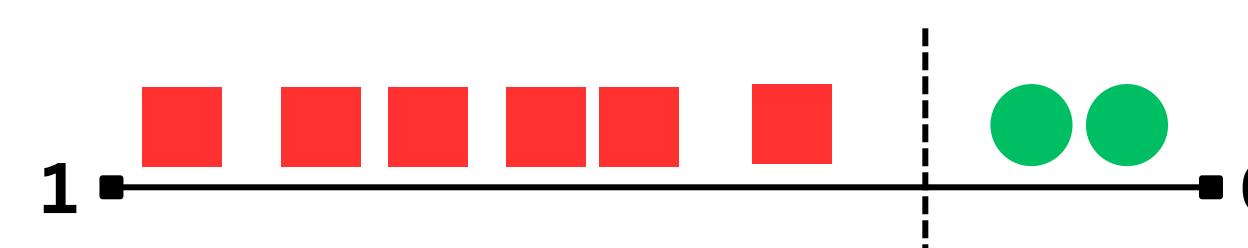
- Ketika Dataset Sangat Tidak Seimbang:**

PR curve lebih informatif ketika dataset sangat tidak seimbang, yaitu ketika kelas positif jauh lebih sedikit dibandingkan kelas negatif. Ini karena PR curve fokus pada presisi (berapa banyak dari hasil positif yang benar-benar positif) dan recall (berapa banyak dari semua positif yang ditemukan), yang lebih relevan dalam konteks ini.

- Ketika Kelas Positif Lebih Penting:**

Jika tujuan utama adalah mengidentifikasi sebanyak mungkin kelas positif dengan mengorbankan beberapa false positives, PR curve akan lebih tepat karena langsung menunjukkan trade-off antara precision dan recall.

Keyword : Hanya memperhatikan **Class Positive**



# Multiclass Confusion Matrix

POV : Apel

	Aktual: Apel	Aktual: Jeruk	Aktual: Pisang
Prediksi: Apel	5	1	0
Prediksi: Jeruk	2	6	1
Prediksi: Pisang	1	1	3

	Aktual: Apel	Aktual: Jeruk	Aktual: Pisang
Prediksi: Apel	TP (5)	FP (1)	FP (0)
Prediksi: Jeruk	FN (2)	TN (6)	TN (1)
Prediksi: Pisang	FN (1)	TN (1)	TN (3)

# **UNSUPERVISED METRICS**

# Silhouette Score

Mengukur seberapa mirip objek dalam satu cluster dengan objek dalam cluster lain. Nilai berkisar antara -1 dan 1, di mana nilai lebih tinggi menunjukkan cluster yang lebih baik.

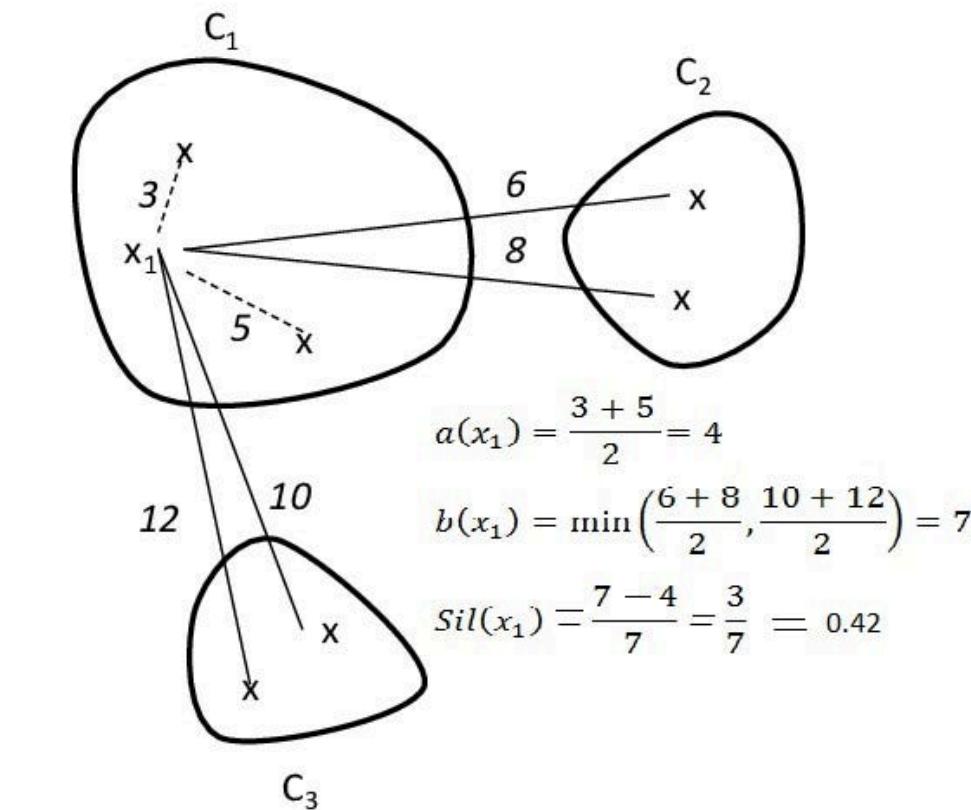
$$\text{Silhouette Score} = \frac{b - a}{\max(a, b)}$$

Di mana  $a$  adalah jarak rata-rata ke titik lain dalam cluster yang sama, dan  $b$  adalah jarak rata-rata ke titik dalam cluster terdekat yang berbeda.

Bayangkan kamu sedang makan malam di restoran dengan beberapa teman, dan ada banyak meja lain di sekitar kamu.

## Dalam Satu Meja (Intra-cluster = $a$ ):

Bayangkan kamu duduk bersama teman-teman dekatmu di satu meja. Kalian semua saling mengenal dan merasa nyaman bersama, sehingga suasana di meja terasa hangat dan menyenangkan.



## Antar Meja (Inter-cluster = $b$ ):

Di meja sebelah, ada sekelompok orang yang tidak kamu kenal. Karena mereka duduk jauh darimu dan kamu tidak memiliki hubungan dengan mereka, kamu tidak terganggu oleh percakapan mereka.

**Silhouette Score mengukur seberapa dekat hubunganmu dengan teman-temanmu di satu meja, dan seberapa jauh hubunganmu dengan orang lain di meja yang berbeda**

# Davies-Bouldin Index

Davies-Bouldin Index (DBI) adalah metrik yang digunakan untuk mengevaluasi kualitas clustering. Metrik ini mengukur seberapa baik cluster yang terbentuk dipisahkan satu sama lain (separation) dan seberapa kompak data dalam masing-masing cluster (compactness). Semakin rendah nilai DBI, semakin baik kualitas clustering.

Bagaimana Davies-Bouldin Index Bekerja:

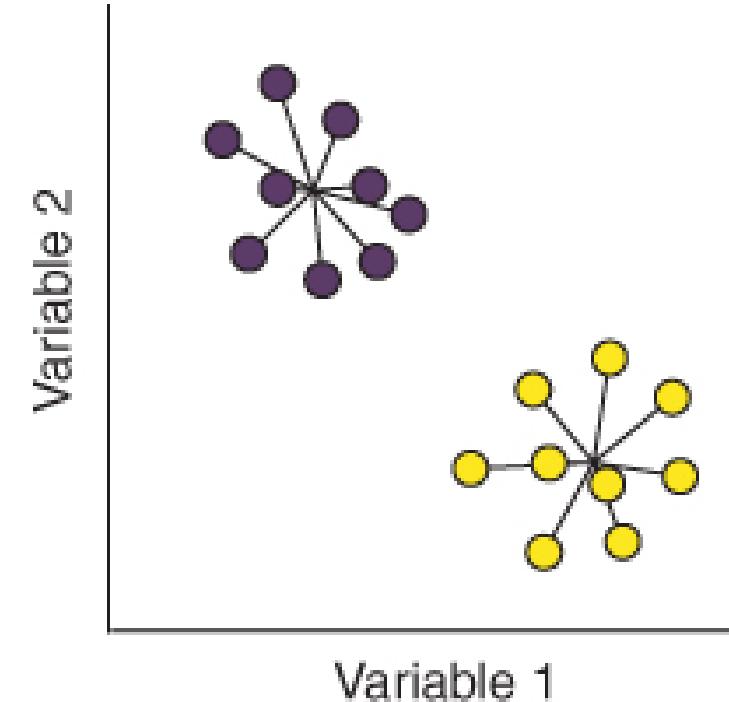
- **Compactness (Kekompakan):** Mengukur seberapa dekat atau rapat data dalam sebuah cluster. Cluster yang baik memiliki data yang rapat.
- **Separation (Pemisahan):** Mengukur seberapa jauh cluster yang satu dengan cluster lainnya. Cluster yang baik memiliki jarak yang jauh dari cluster lainnya.

Bayangkan kamu seorang pemilik restoran dan sedang mengamati setiap meja makan para pengunjung restoran.

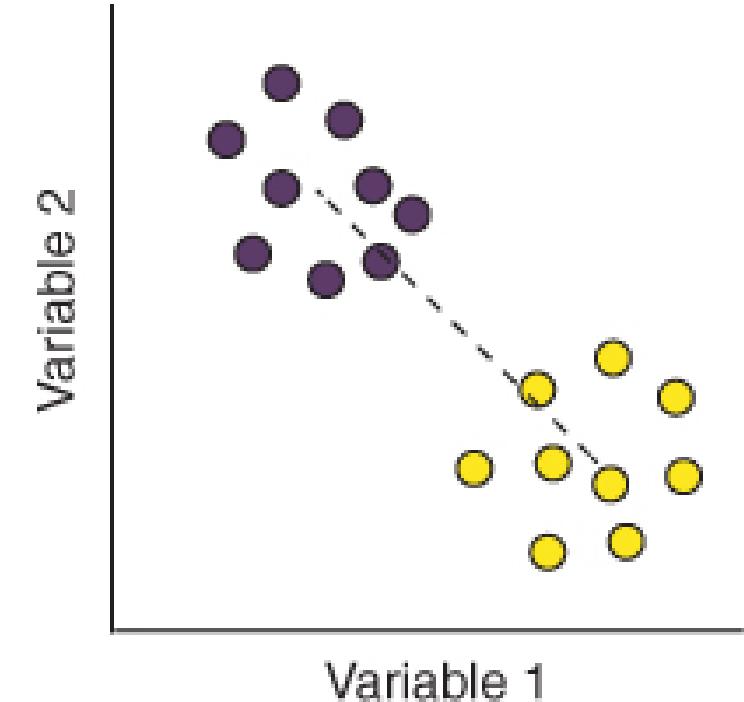
## Dalam Satu Meja (Intra-cluster):

Kamu melihat beberapa meja yang isinya hanya diisi satu keluarga, ada satu meja yang hanya diisi orang-orang satu kantor, ada satu meja yang diisi oleh sekumpulan mahasiswa dari jurusan yang sama

## Intracluster variance



## Distance between centroids



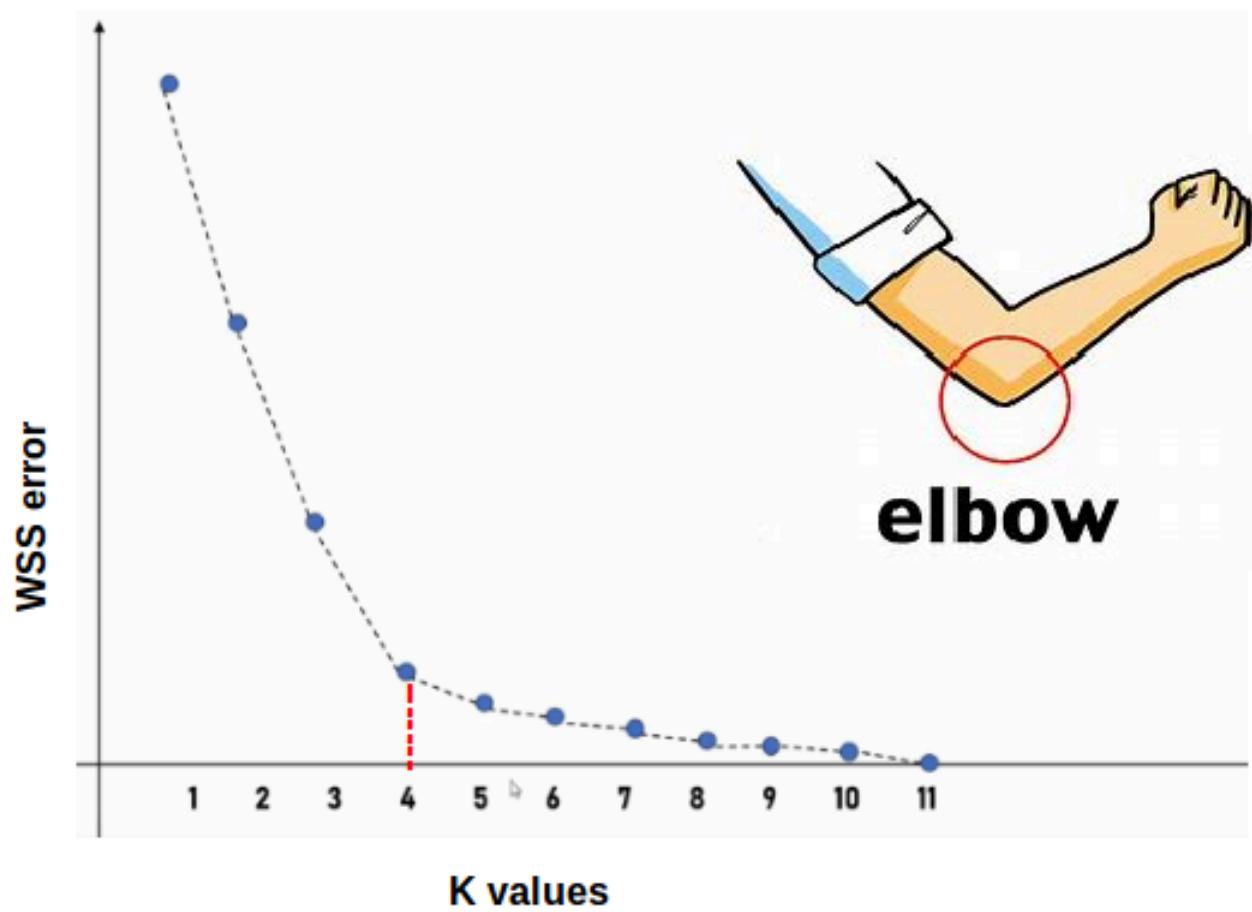
## Antar Meja (Inter-cluster):

Kamu melihat antar meja tidak ada interaksi karena pengunjung tersebut tidak saling kenal satu sama lain diluar meja mereka masing-masing

**Davies-Bouldin Index mengukur seberapa dekat orang-orang di satu meja yang sama, dan seberapa jauh hubungan antara meja satu dengan meja lainnya**

# Elbow Method

Elbow Method adalah teknik yang digunakan untuk menentukan jumlah optimal cluster dalam algoritma clustering seperti K-Means. Metode ini bekerja dengan menghitung total within-cluster sum of squares (WCSS) untuk sejumlah nilai k (jumlah cluster) dan memilih k di mana penurunan WCSS mulai melambat, membentuk "elbow" atau titik bahu pada grafik.

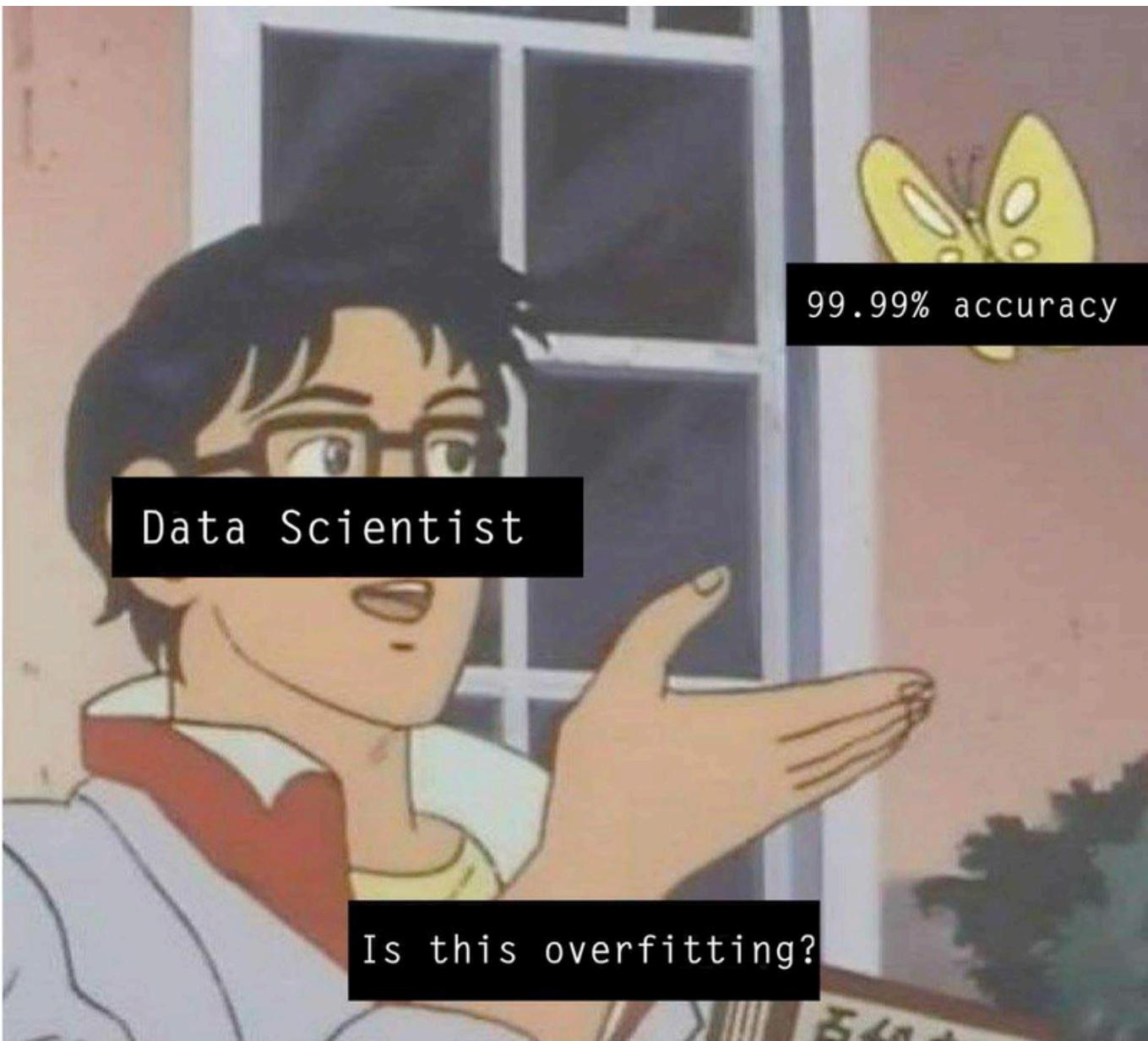


Misal kamu membuat pesta makan malam dan kamu harus menentukan berapa meja untuk teman-temanmu dan memisahkan mereka supaya mereka satu meja dengan teman yang lebih cocok satu sama lain

- **1 Meja:** Semua teman duduk di satu meja besar. Tapi, percakapan jadi berisik dan tidak semua orang bisa berbicara dengan nyaman.
- **2 Meja:** Kamu membagi teman-teman menjadi dua kelompok. Percakapan jadi lebih tenang, tapi masih ada beberapa teman yang merasa kurang cocok dengan orang di meja mereka.
- **3 Meja:** Sekarang, percakapan di setiap meja lebih cocok karena kamu mengelompokkan teman-teman dengan minat yang mirip.
- **4 Meja:** Meja-meja terasa nyaman, tapi kalau kamu menambahkan meja kelima, pembagiannya sudah tidak membuat banyak perbedaan lagi.

# OPTIMIZATION

# Why Optimization so Important?



Apakah model dengan akurasi yang sangat tinggi itu adalah model yang sangat bagus?

Model dikatakan bagus jika akurasi pada model semakin tinggi, tapi ketika sangat tinggi justru bisa bisa menimbulkan masalah bernama **OVERFITTING**

# OVERFITTING

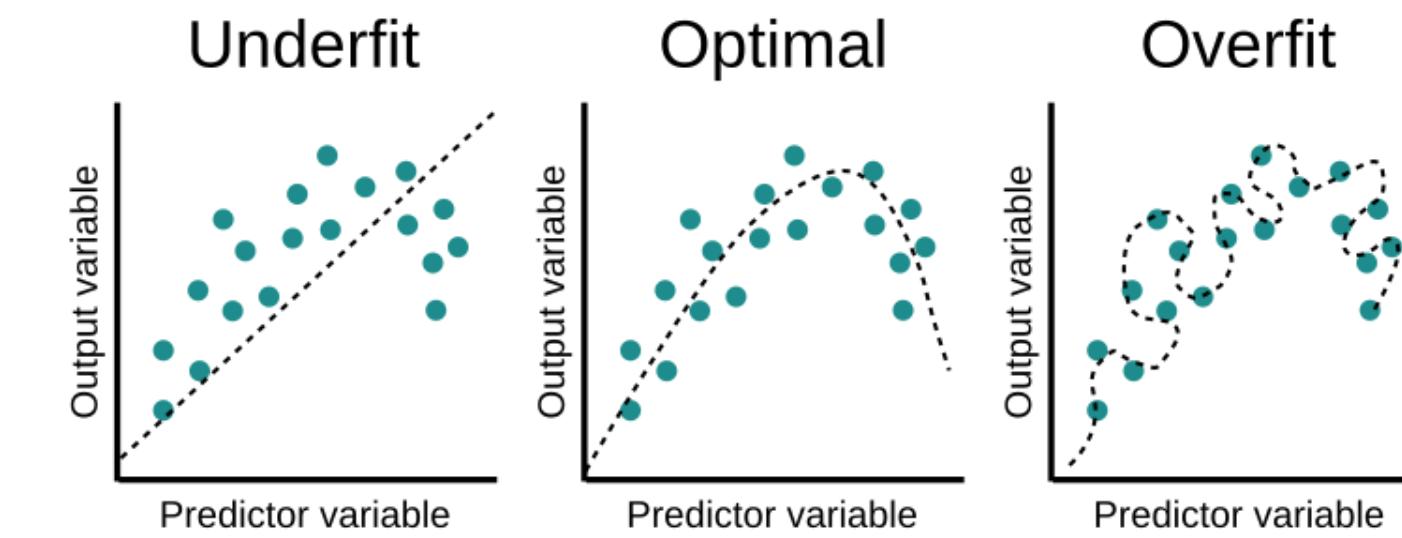
## Model on Training



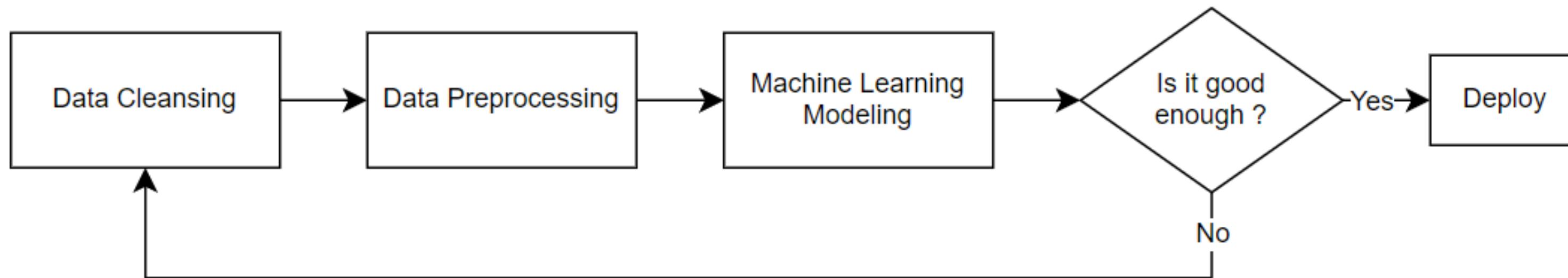
## Model on Testing



Salah satu ciri model mengalami overfitting adalah ketika akurasi pada saat Testing lebih buruk daripada akurasi pada saat Training, karena model terlalu mempelajari pola data Training



# OPTIMIZATION



## Mengulang tahapan Cleansing dan Preprocessing

- Perhatikan kembali apakah data sudah balance
- Perhatikan apakah masih terdapat outlier
- Perhatikan korelasi feature terhadap target
- Lakukan kembali feature engineering
- Lakukan evaluasi lebih dalam

# Hyperparameter Tuning

Hyperparameter tuning adalah proses untuk mengoptimalkan hyperparameter dalam model machine learning. Hyperparameter adalah parameter yang harus ditentukan sebelum pelatihan model dimulai dan tidak dipelajari dari data. Contoh hyperparameter adalah jumlah pohon dalam Random Forest, learning rate dalam Gradient Boosting, atau jumlah Neighbour dalam K-Nearest Neighbors (KNN).

```
from xgboost import XGBRegressor

model = XGBRegressor(learning_rate=0.5, n_estimators=100, max_depth=4)
```

## Hyperparameter Tuning Using Cross Validation

```
rf = RandomForestRegressor(random_state=42)

param_dist = {
    'n_estimators': randint(50, 200),
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': randint(2, 11),
    'min_samples_leaf': randint(1, 5)
}

random_search = RandomizedSearchCV(estimator=rf,
                                    param_distributions=param_dist,
                                    n_iter=50, cv=5,
                                    n_jobs=-1, random_state=42,
                                    scoring='neg_mean_squared_error')
random_search.fit(X_train, y_train)
```

### Random Search

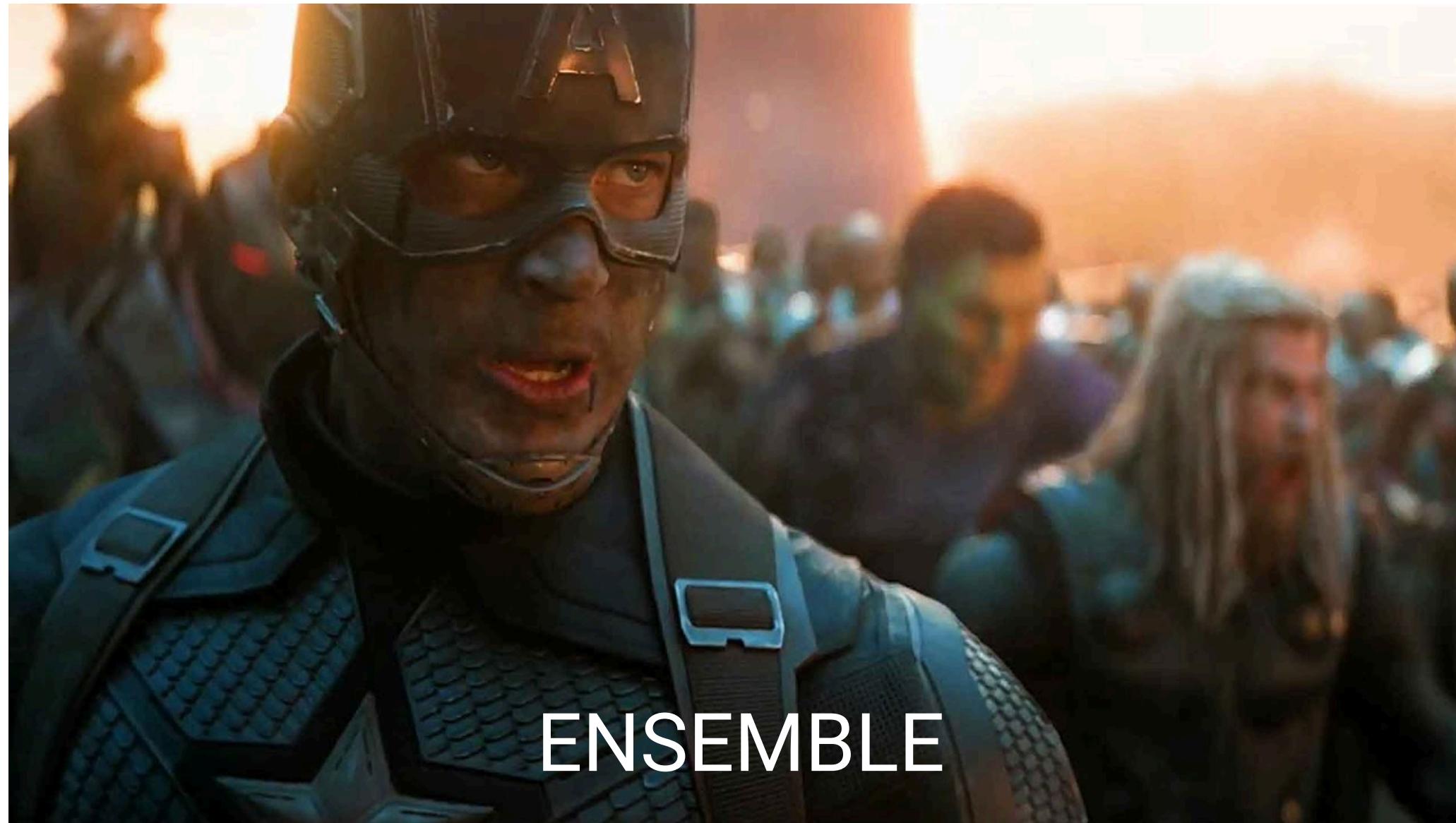
```
rf = RandomForestRegressor(random_state=42)

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

grid_search = GridSearchCV(estimator=rf,
                           param_grid=param_grid,
                           cv=5, n_jobs=-1,
                           scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)
```

### Grid Search

# Ensembling

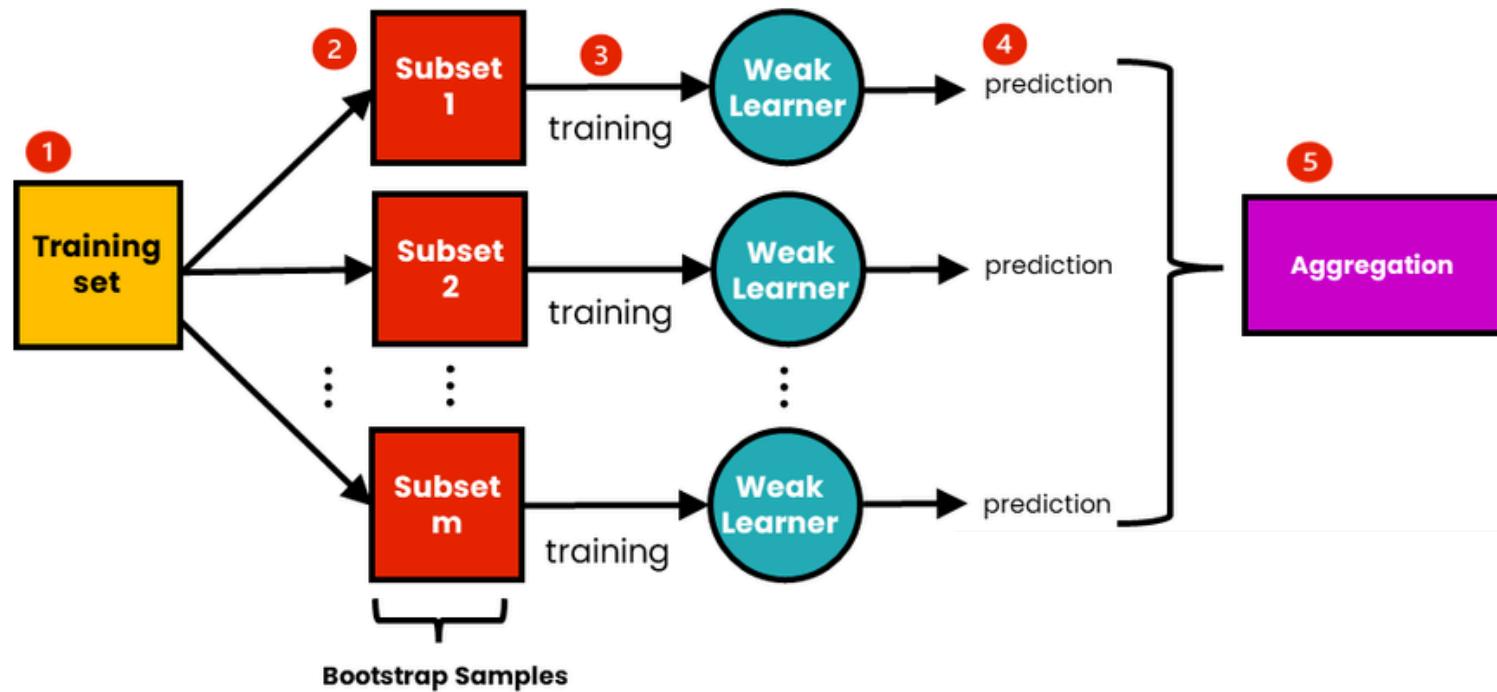


# Ensembling

**Ensembling** adalah teknik dalam machine learning yang **menggabungkan beberapa model untuk menghasilkan prediksi yang lebih kuat dan lebih akurat daripada model tunggal**. Ideanya adalah dengan menggabungkan prediksi dari beberapa model, kesalahan yang dibuat oleh model individual dapat dikompensasi oleh model lain, sehingga meningkatkan akurasi dan stabilitas prediksi secara keseluruhan.

- **Bagging**

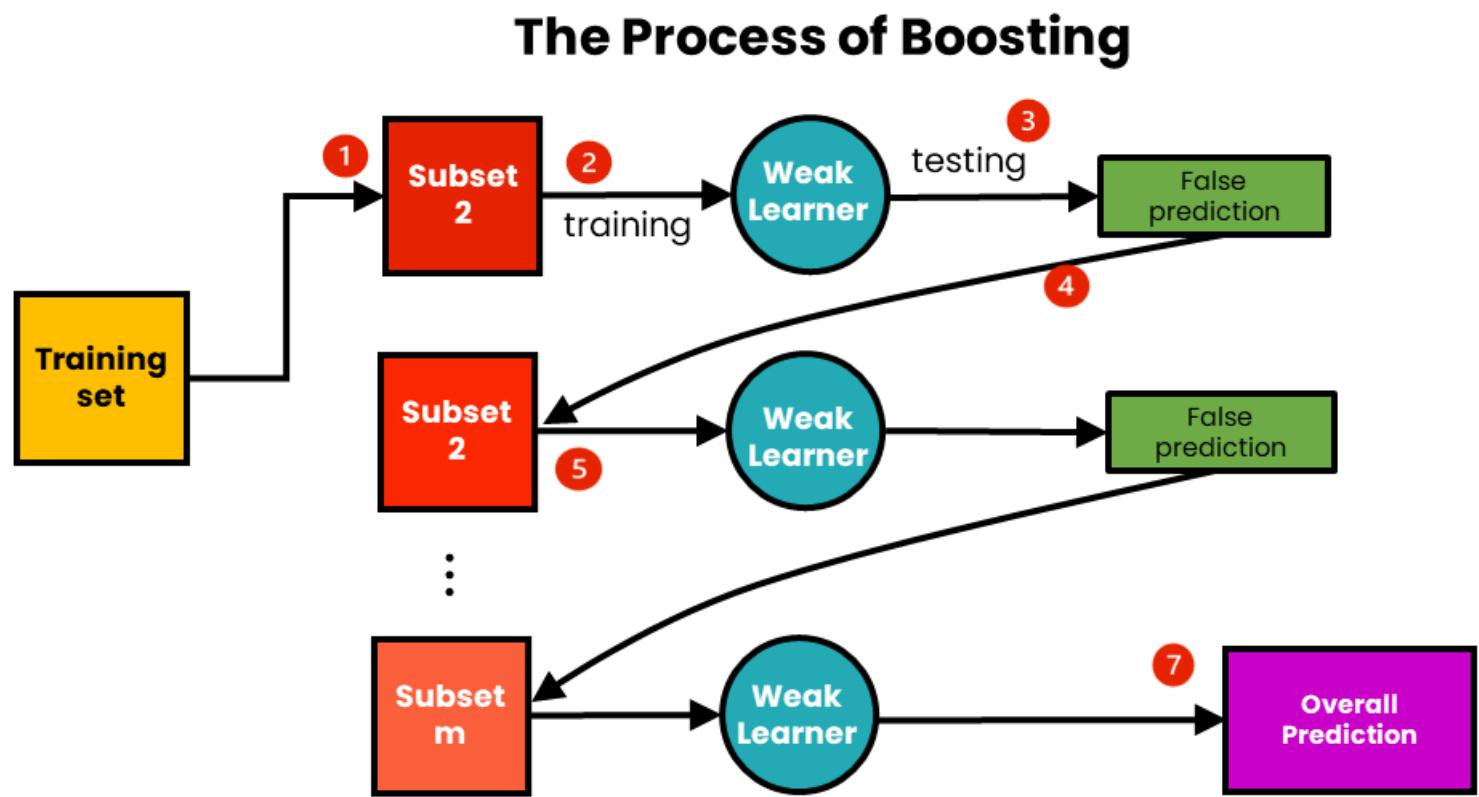
**The Process of Bagging (Bootstrap Aggregation)**



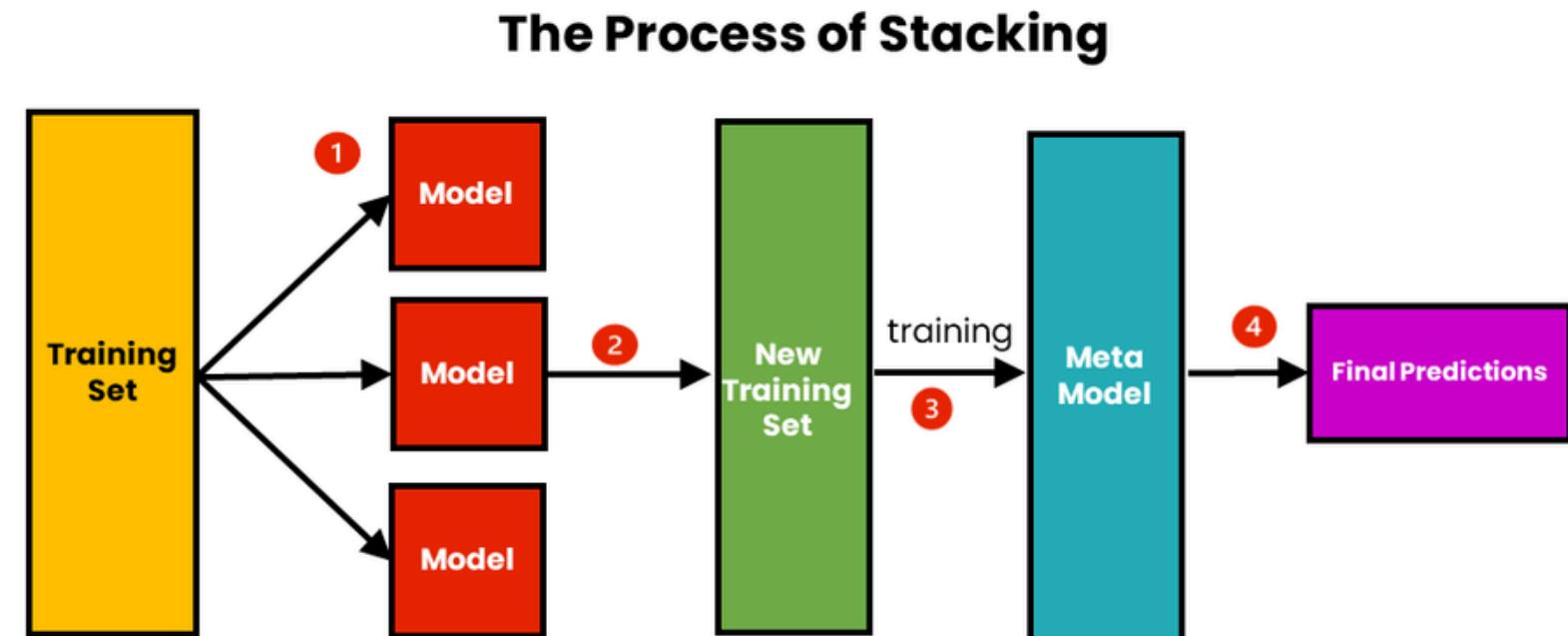
Bagging adalah teknik ensembling yang melatih beberapa model independen secara paralel **menggunakan subset data yang berbeda** dan menggabungkan hasilnya dengan cara rata-rata (untuk regresi) atau voting mayoritas (untuk klasifikasi).

Contoh : **RandomForrest**

## • Boosting



## • Stacking



**Boosting** adalah teknik ensembling yang melatih model secara berurutan. **Setiap model baru mencoba memperbaiki kesalahan model sebelumnya.** Model individu dalam boosting biasanya lemah (weak learners)

Contoh : **Gradient Boosting**

**Stacking** menggabungkan prediksi dari beberapa model berbeda (bisa jadi model yang sama atau berbeda jenis) dengan **menggunakan model lain yang disebut sebagai "meta-learner" atau "meta-model".** Meta-model ini belajar untuk membuat prediksi berdasarkan output dari model-model sebelumnya.

**WEEK 5**

# **DEPLOYMENT MACHINE LEARNING**

**MENTOR : EJI**

Events Video Special Issues Jobs

VentureBeat



Subscribe

GamesBeat

Artificial Intelligence ▾

Security ▾

Data Infrastructure ▾

Automation ▾

Enterprise Analytics ▾

More ▾

Sponsored

# Why do 87% of data science projects never make it into production?

VB Staff

---

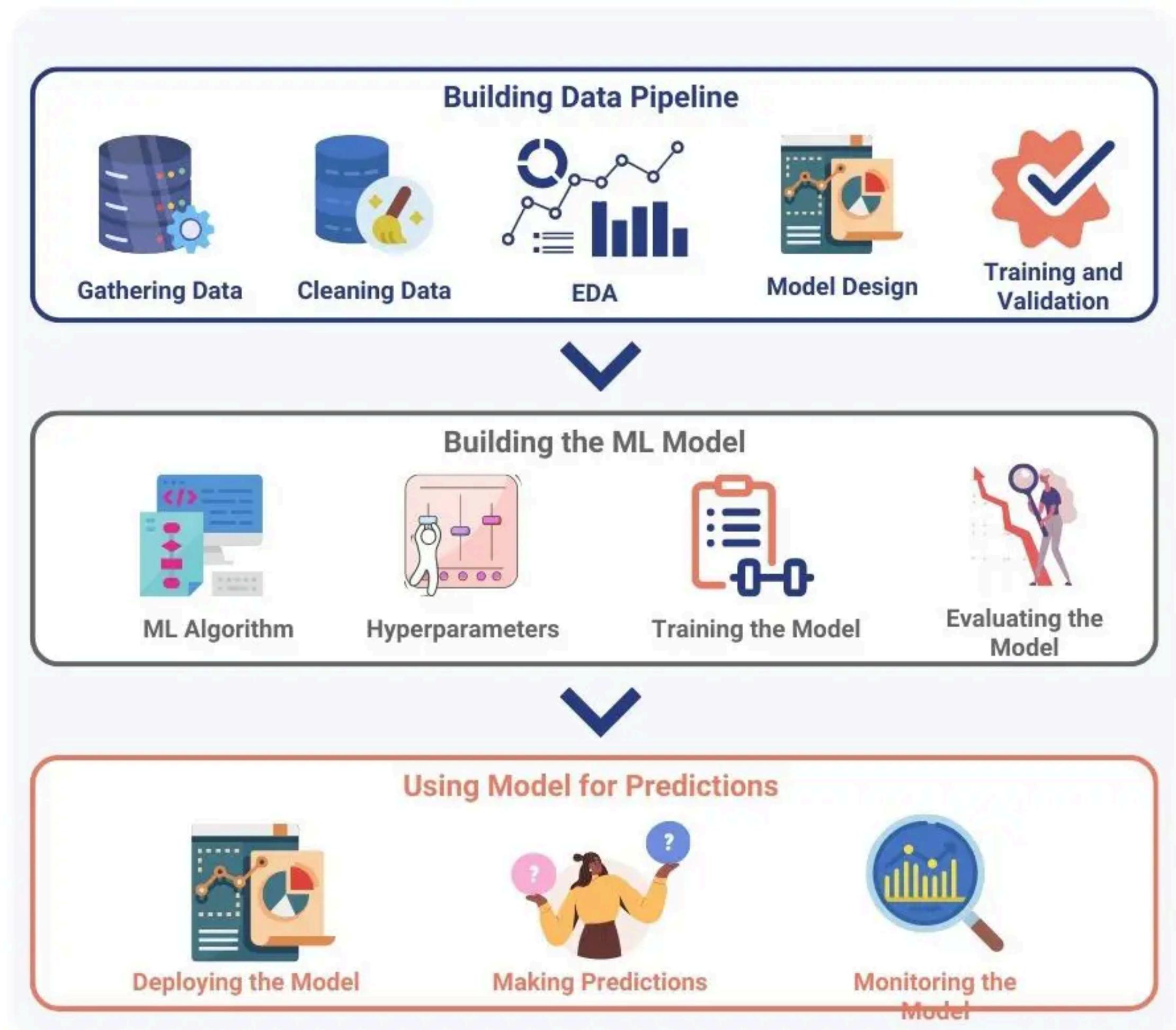
July 19, 2019 4:10 AM

---

f X in

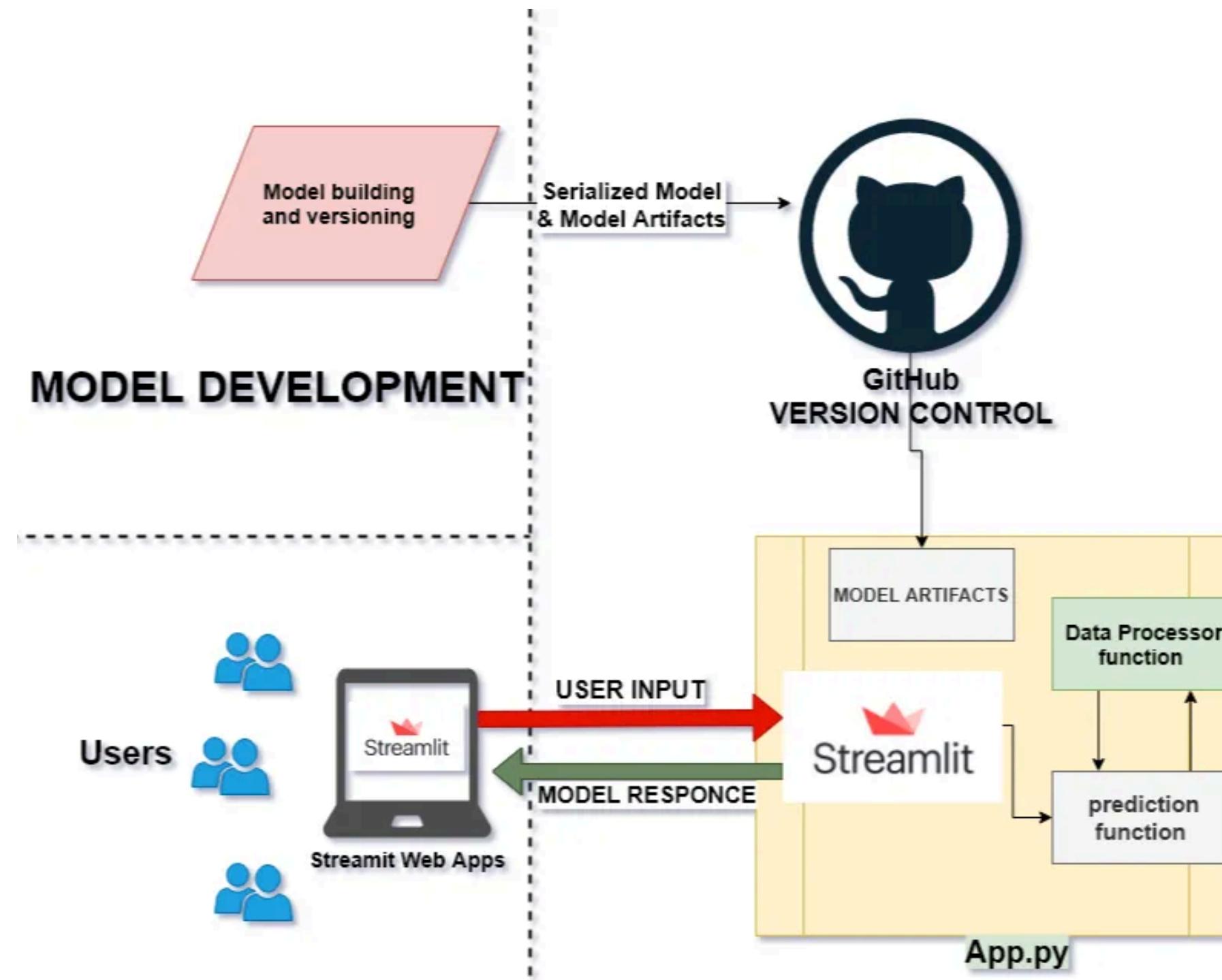


**Model deployment** is the process of making  
a machine learning model **available** for use  
in **production environments**.

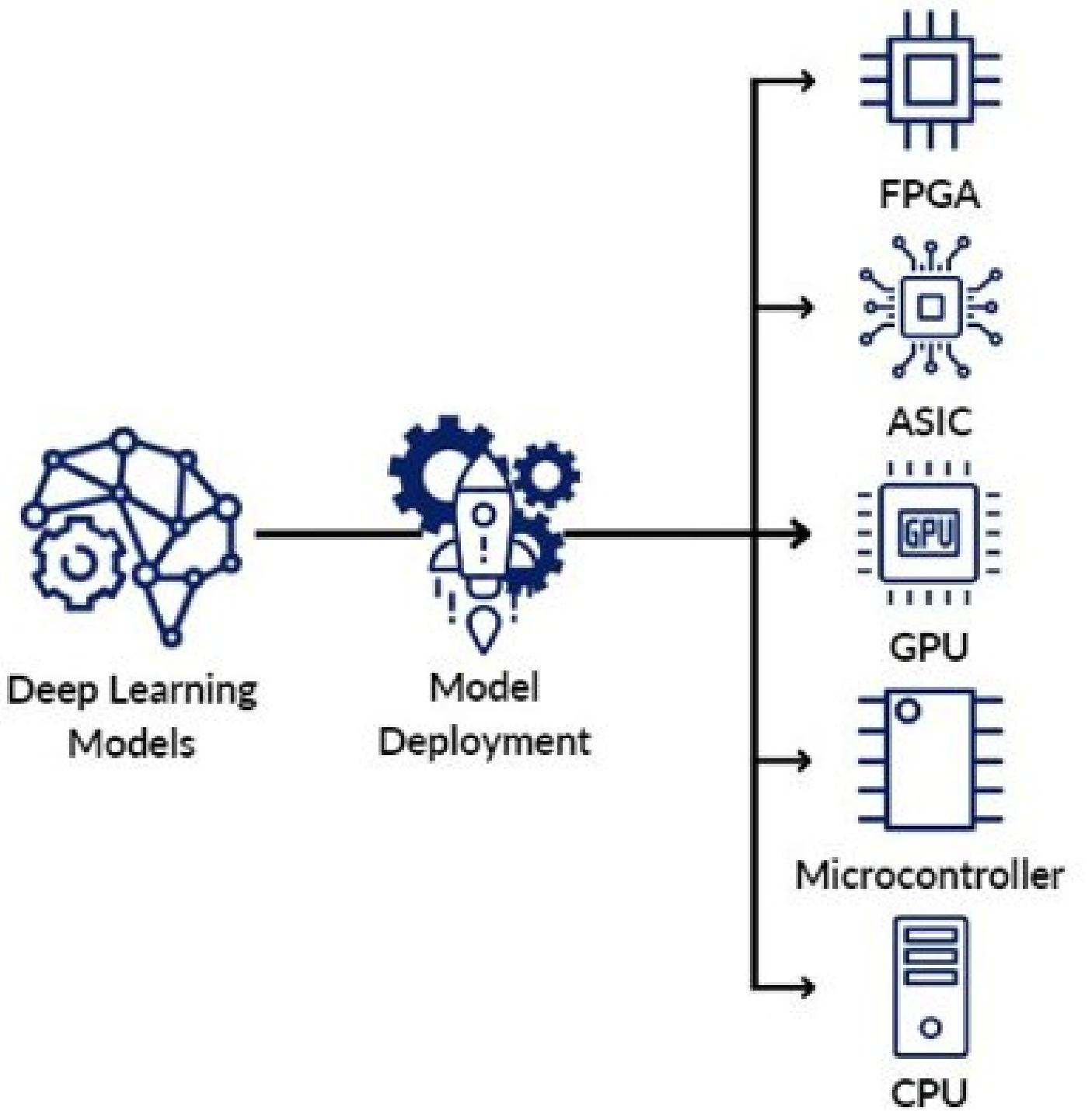


# Metode Deployment

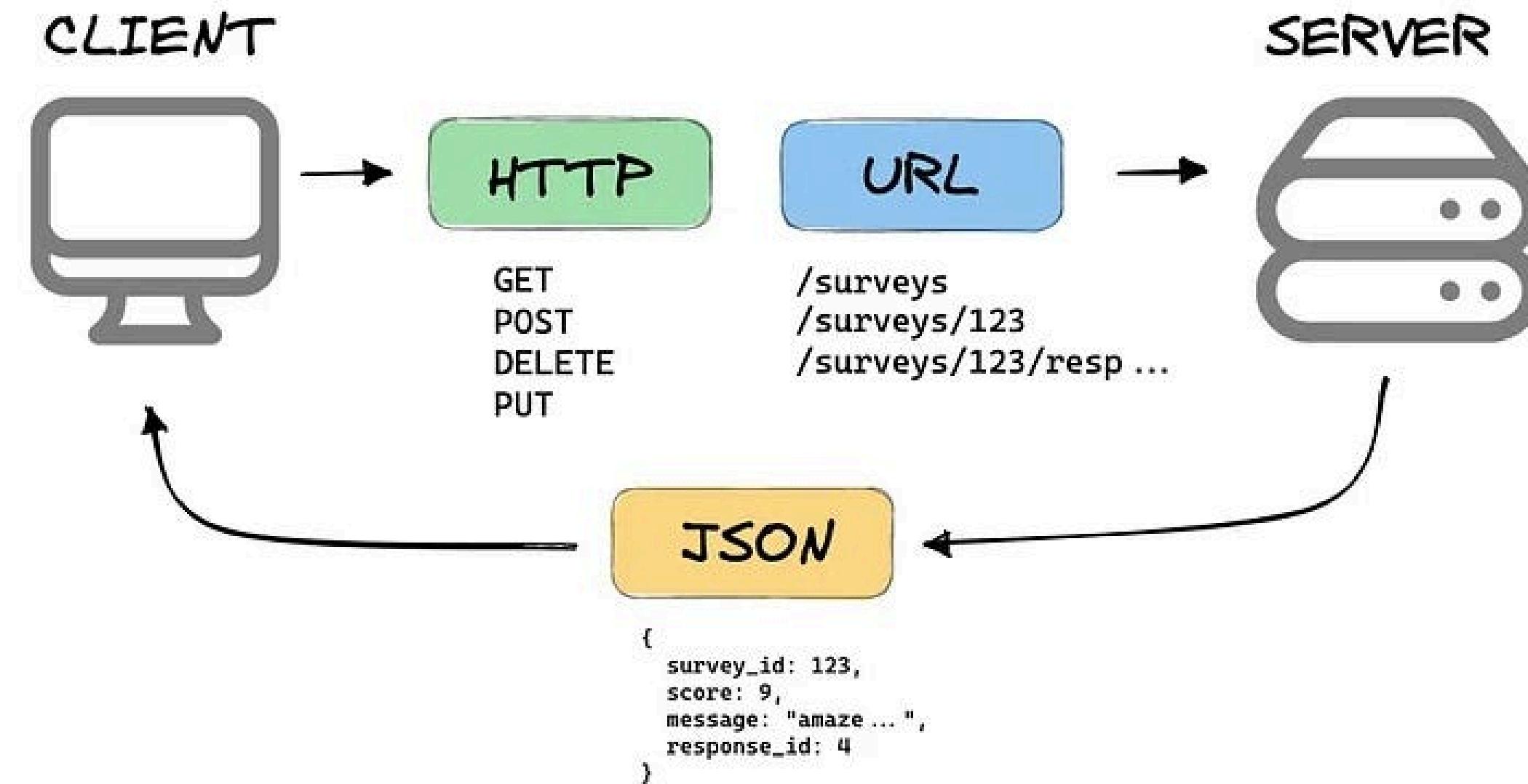
# Real-Time Inference



# Embedded Systems



# REST API



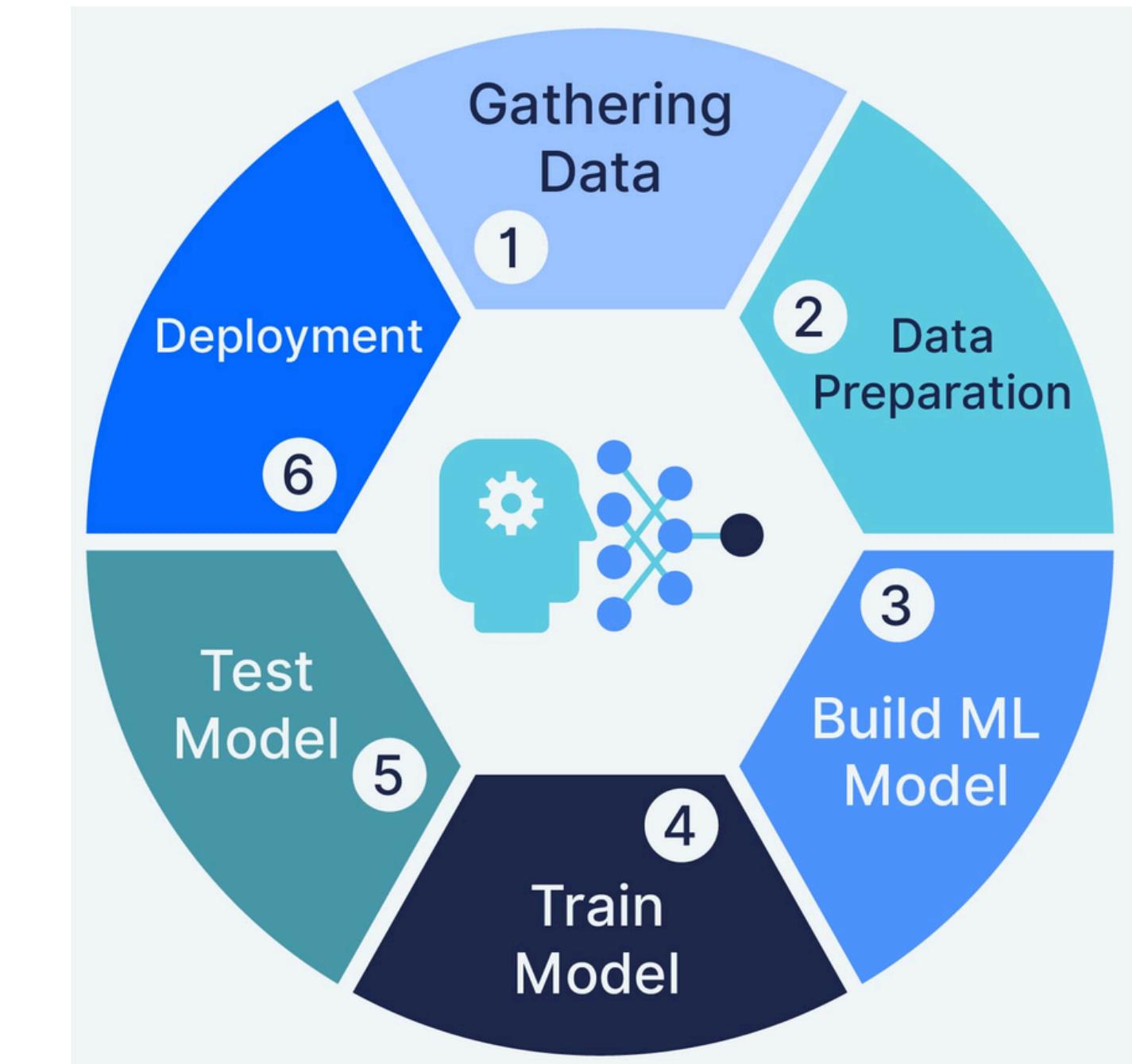
**Lainnya....**

**Batch Inference**

**Microservices Architecture**

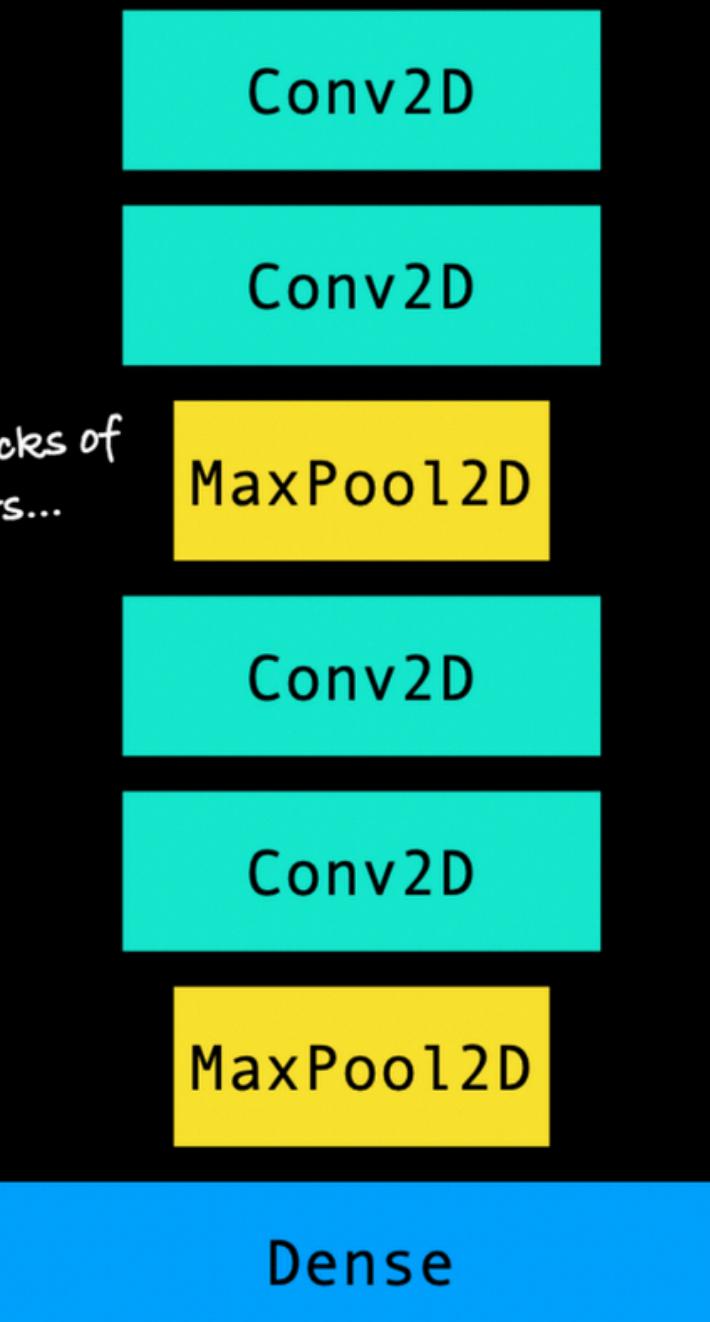
**Containerization**

**Serverless Computing**



# model building == model deployment

Lego blocks of layers...



TensorFlow model on Google Storage



(sort of)

Lego blocks of tools...

Model hosted on AI Platform



App built with Streamlit



App wrapped up in Docker container



Docker container deployed to GCR



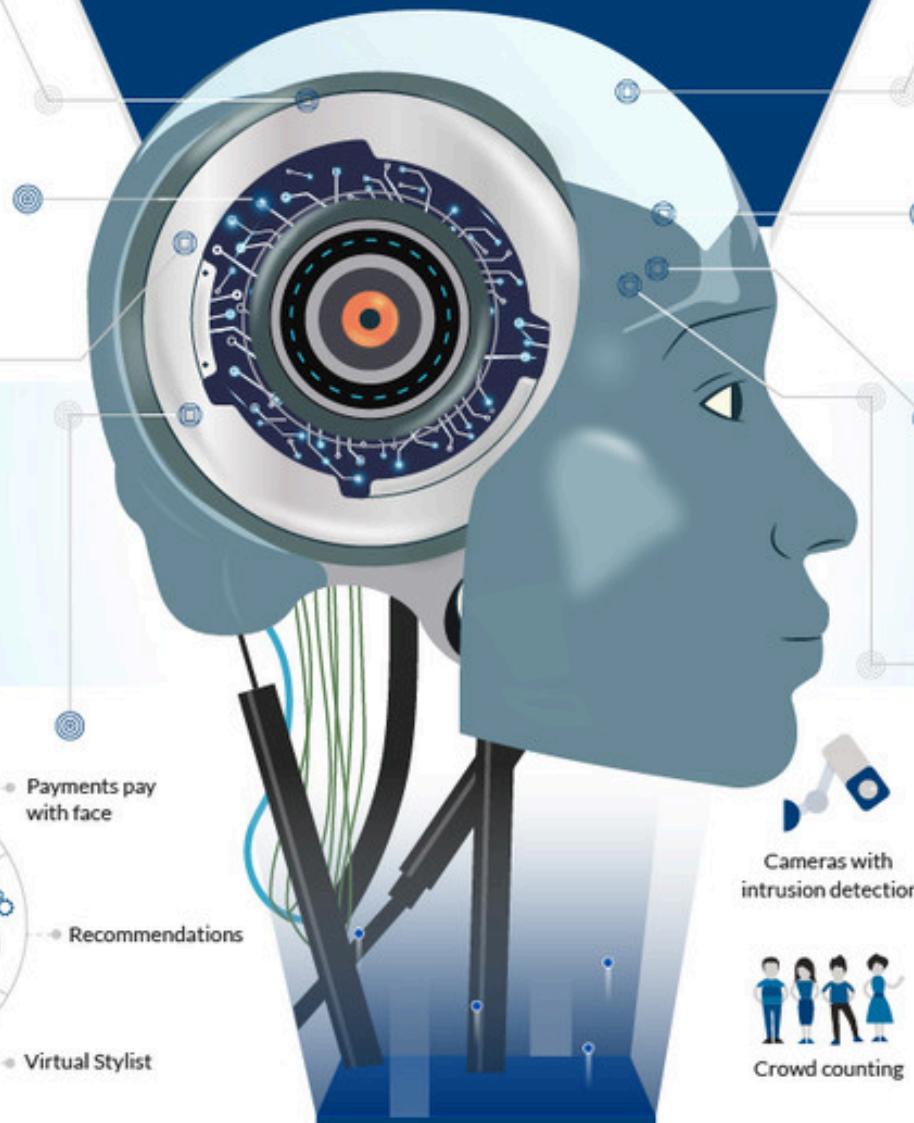
GCR = Google Container Registry

GCR hosted container deployed to App Engine

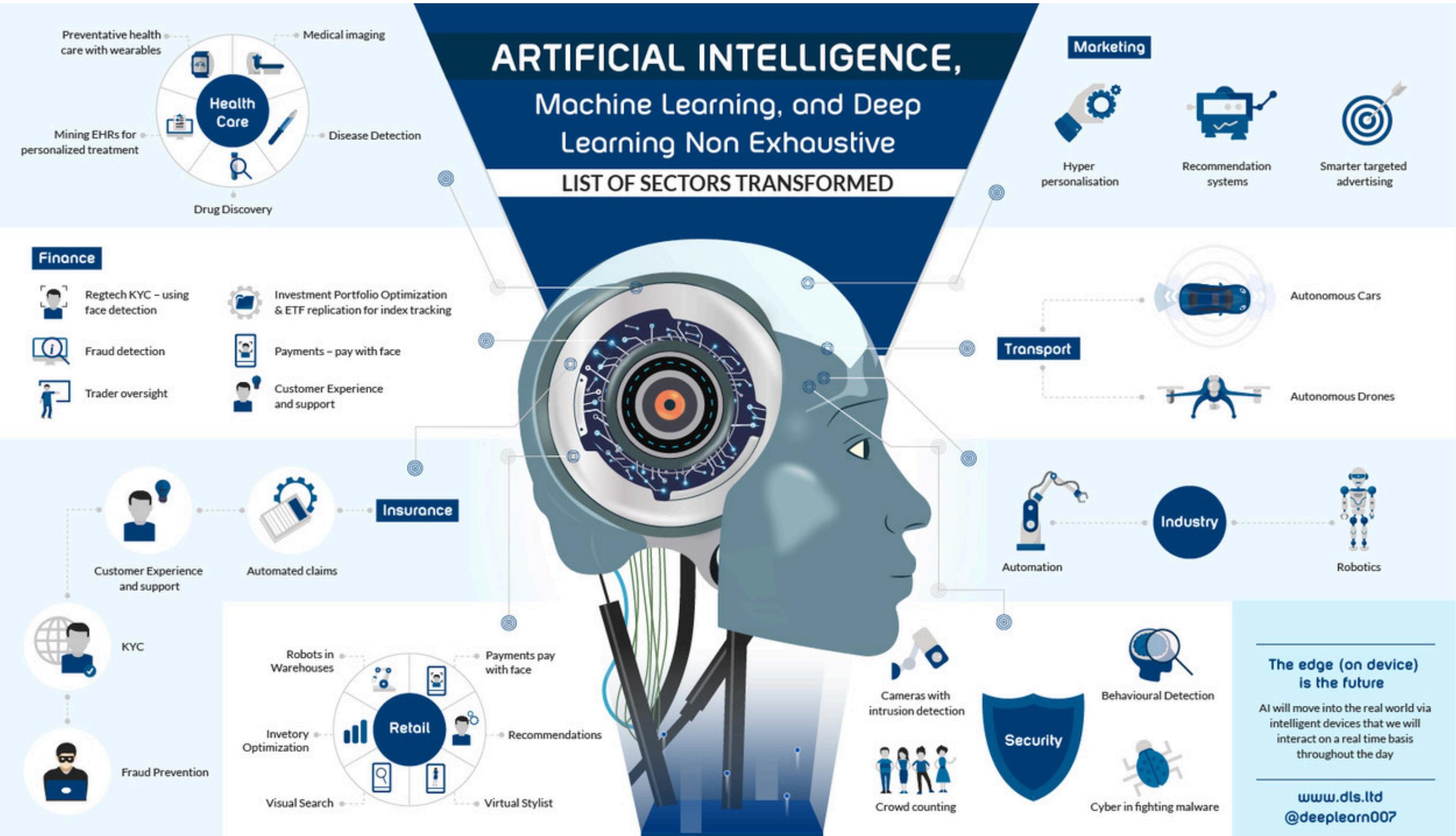


App monitored with Google Monitoring





# ARTIFICIAL INTELLIGENCE, Machine Learning, and Deep Learning Non Exhaustive LIST OF SECTORS TRANSFORMED



## DEPLOYMENT MACHINE LEARNING

### Inputs

sepal length (cm)  
5.00

sepal width (cm)  
3.60

petal length (cm)  
1.40

petal width (cm)  
0.20

## Iris Flower Classification

This app correctly classifies iris flower among 3 possible species

### Results

Following is the probability of each class

	setosa	versicolor	virginica
result	0.9794	0.0206	0

This flower belongs to setosa class

