DigitalSkola
Uncover The World of Digital Skills with Us

# Day 2: Exploratory Data Analysis

## Hands-on how to interpret data into visualization

# Table of Content

## What will We Learn Today?

1. Definisi *Univariate* dan *Bivariate* analisis
2. Membuat visualisasi (bar chart, line chart, pie chart, stacked bar chart, boxplot, heatmap correlation)
3. Custom visualisasi + kumpulan *function*
4. Membaca insight dari visualisasi

# Mengapa Visualisasi diperlukan sebelum Membuat Model Machine Learning?

- Memahami Struktur dan Karakteristik Data
- Menemukan Pola, Tren, dan Hubungan
- Mengidentifikasi Masalah Data Sejak Awal
- Membantu dalam Feature Selection
- Membuat Model Lebih Transparan dan Bisa Dipertanggungjawabkan

# Univariate Analysis

Analisis terhadap **satu kolom/fitur saja**, untuk memahami karakteristik data tersebut.

Contoh:

- Melihat **distribusi usia** karyawan.
- Mengetahui **berapa banyak** karyawan yang dipromosikan.

🛠️ **Visualisasi yang digunakan**:

- Bar chart (untuk data kategorikal)
- Histogram / Boxplot (untuk data numerik)

# Bivariate Analysis

Analisis **hubungan antara dua variabel**, biasanya antara fitur dan target.

Contoh:

- Apakah usia berpengaruh terhadap peluang promosi?
- Apakah jenis kelamin berpengaruh terhadap gaji?

🛠️ **Visualisasi yang digunakan**:

- Scatter plot (numerik vs numerik)
- Heatmap korelasi (numerik vs numerik dalam jumlah besar)

# Tools Data Visualisasi

Beberapa package/library Python




#BertalentaDigital


# Membuat Visualisasi Di Python

# Perlu diingat!!!

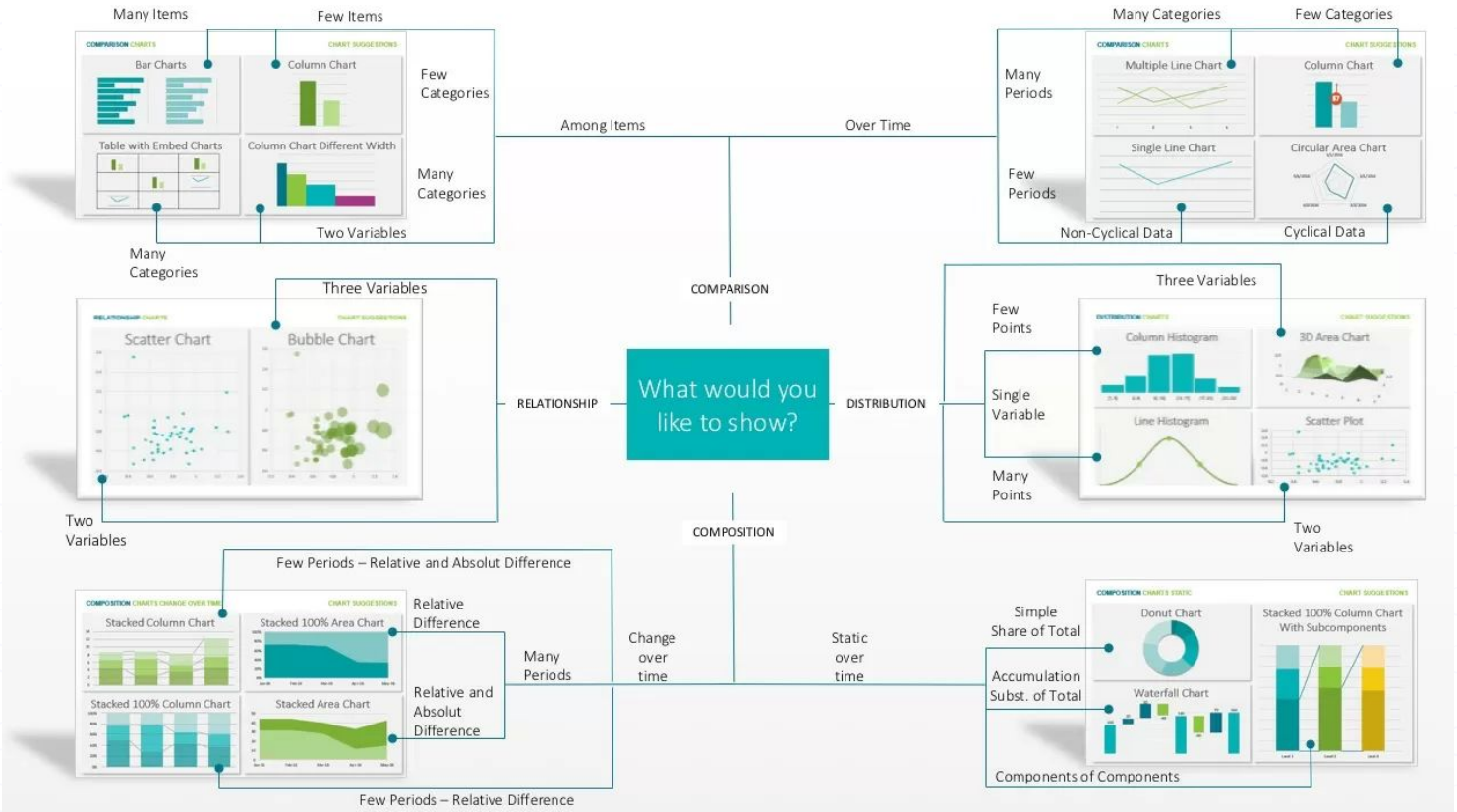
**30%**
Mempresentasikan insight dari visualisasi dan *action-item*

**50%**
Pahami dahulu challenge/problem akan impact bisnis

**10%**
Ngoding visualisasi untuk melihat tren/hubungan dari data

**10%**
Memilah visualisasi sesuai relevansi ke impact bisnis

# Starter Chart



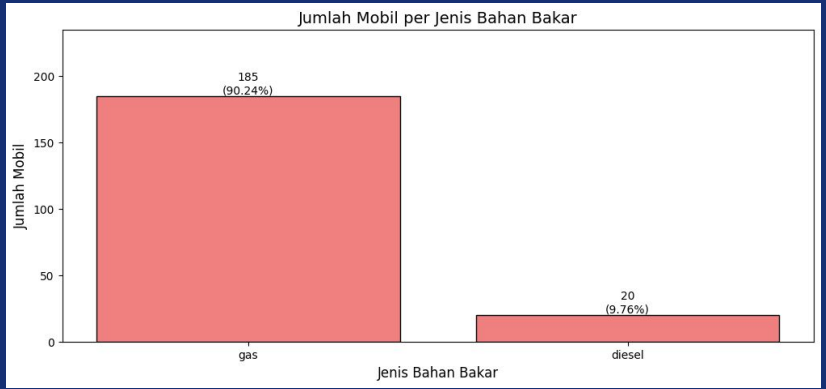
# Bar Chart

Struktur data sebagai berikut:

| | fueltype | CarName |
|---|---|---|
| 1 | gas | 185 |
| 0 | diesel | 20 |

```
cus_viz.simple_pivot(data=data, ind='fueltype',
val='CarName', agg='count')
```

**Catatan:** struktur data dan hasil visualisasi menggunakan function dari Python file collection_function.py



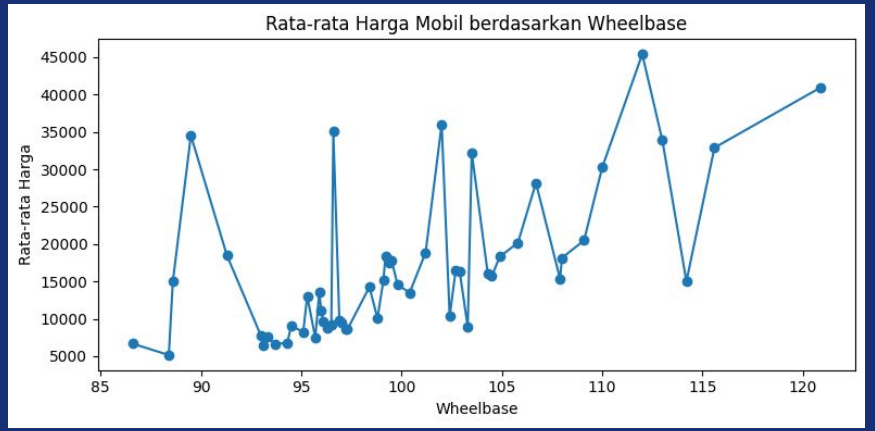
# Line Chart

Struktur data sebagai berikut:

| | wheelbase | price |
|---|---|---|
| 0 | 86.6 | 6667.000000 |
| 1 | 88.4 | 5151.000000 |
| 2 | 88.6 | 14997.500000 |
| 3 | 89.5 | 34528.000000 |

```
cus_viz.simple_pivot(data=data,
ind='wheelbase', val='price', agg='mean')
```

**Catatan:** struktur data dan hasil visualisasi menggunakan function dari Python file collection_function.py



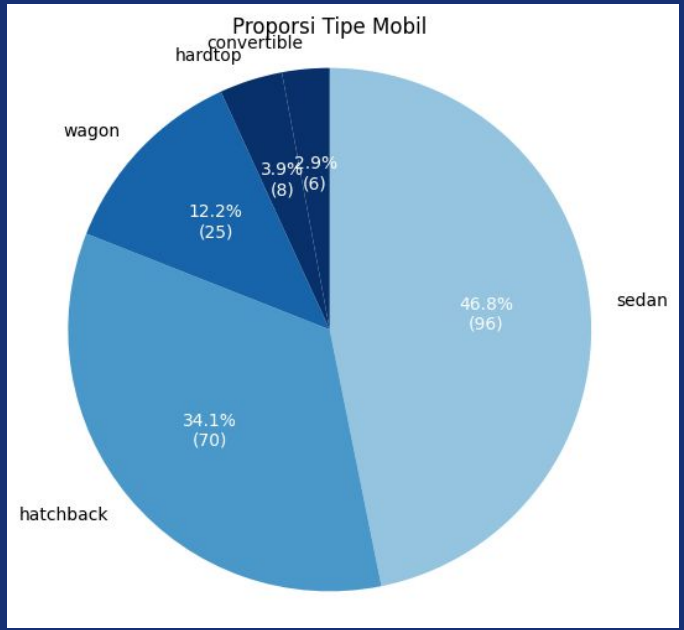
# Pie Chart

Struktur data sebagai berikut:

| | carbody | CarName |
|---|---|---|
| 3 | sedan | 96 |
| 2 | hatchback | 70 |
| 4 | wagon | 25 |
| 1 | hardtop | 8 |
| 0 | convertible | 6 |

```
cus_viz.simple_pivot(data=data,
ind='carbody', val='CarName', agg='count')
```

**Catatan:** struktur data dan hasil visualisasi menggunakan function dari Python file collection_function.py

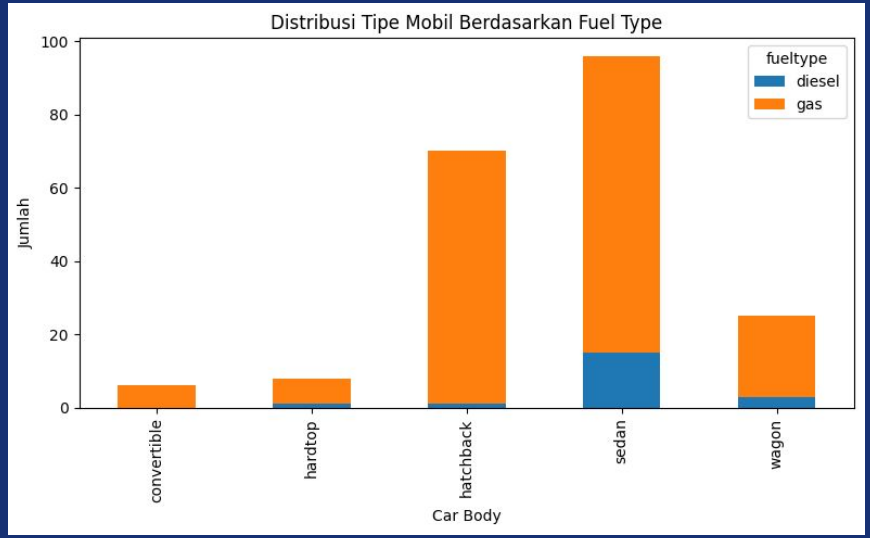


# Stacked Bar Chart

Struktur data sebagai berikut:

| fueltype<br>carbody | diesel | gas |
|---|---|---|
| convertible | 0 | 6 |
| hardtop | 1 | 7 |
| hatchback | 1 | 69 |
| sedan | 15 | 81 |
| wagon | 3 | 22 |

```
pd.crosstab(data['carbody'], data['fueltype'])
```

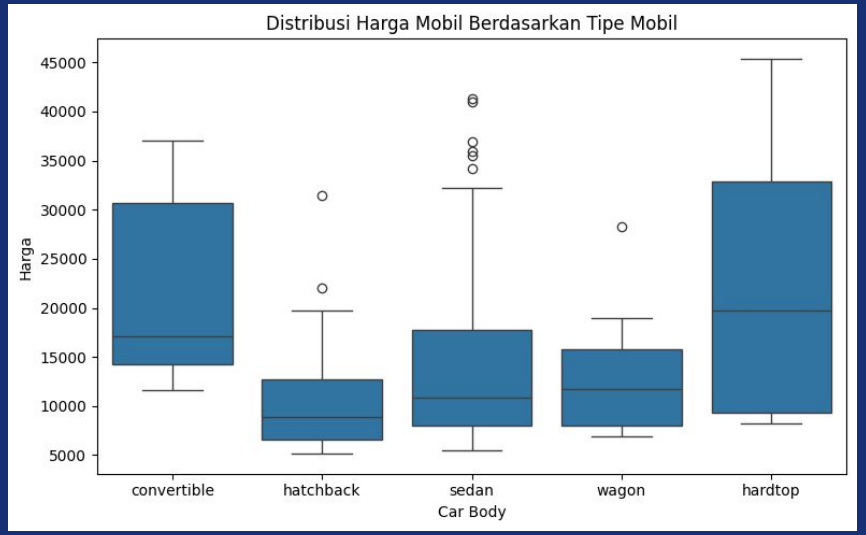


# Boxplot Chart

Struktur data sebagai berikut:

```
sns.boxplot(data=data, x='carbody', y='price')
```

**Catatan:** boxplot tidak memerlukan transformasi data, data original saja untuk menampilkan distribusi dari suatu data.

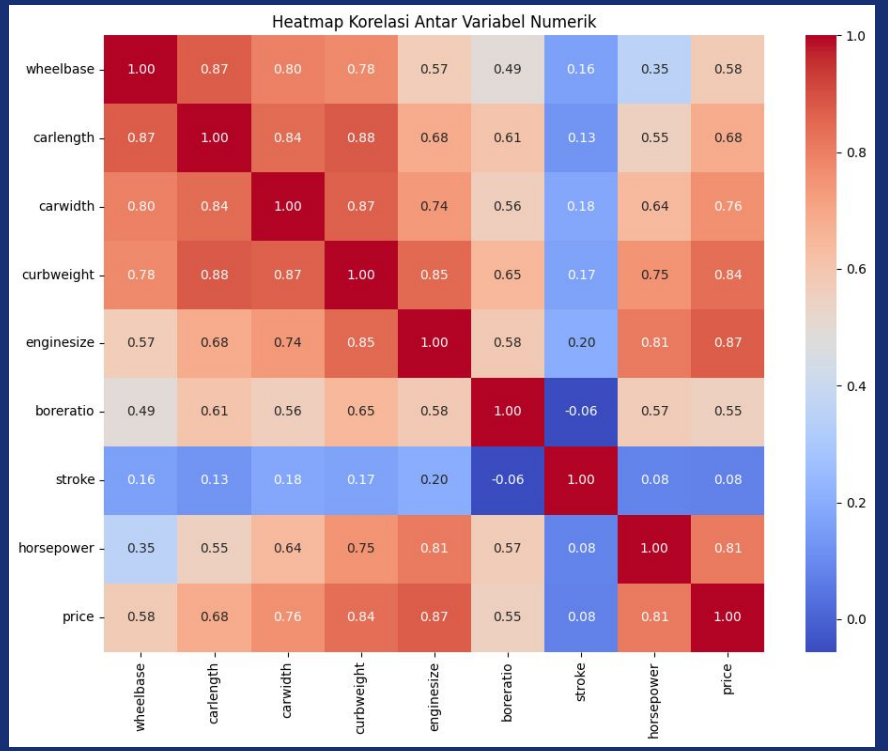


# Heatmap Correlation

Struktur data sebagai berikut:

| | wheelbase | carlength | carwidth | curbweight | enginesize | boreratio | stroke | horsepower | price |
|---|---|---|---|---|---|---|---|---|---|
| **wheelbase** | 1.000000 | 0.874587 | 0.795144 | 0.776386 | 0.569329 | 0.488750 | 0.160959 | 0.353294 | 0.577816 |
| **carlength** | 0.874587 | 1.000000 | 0.841118 | 0.877728 | 0.683360 | 0.606454 | 0.129533 | 0.552623 | 0.682920 |
| **carwidth** | 0.795144 | 0.841118 | 1.000000 | 0.867032 | 0.735433 | 0.559150 | 0.182942 | 0.640732 | 0.759325 |
| **curbweight** | 0.776386 | 0.877728 | 0.867032 | 1.000000 | 0.850594 | 0.648480 | 0.168790 | 0.750739 | 0.835305 |
| **enginesize** | 0.569329 | 0.683360 | 0.735433 | 0.850594 | 1.000000 | 0.583774 | 0.203129 | 0.809769 | 0.874145 |
| **boreratio** | 0.488750 | 0.606454 | 0.559150 | 0.648480 | 0.583774 | 1.000000 | -0.055909 | 0.573677 | 0.553173 |
| **stroke** | 0.160959 | 0.129533 | 0.182942 | 0.168790 | 0.203129 | -0.055909 | 1.000000 | 0.080940 | 0.079443 |
| **horsepower** | 0.353294 | 0.552623 | 0.640732 | 0.750739 | 0.809769 | 0.573677 | 0.080940 | 1.000000 | 0.808139 |
| **price** | 0.577816 | 0.682920 | 0.759325 | 0.835305 | 0.874145 | 0.553173 | 0.079443 | 0.808139 | 1.000000 |

```
numeric_cols.corr()
```



#BertalentaDigital


# Hands-on Python

Buka Notebook Exploratory Data Analysis

DigitalSkola
Uncover The World of Digital Skills with Us

# Thank You.

DigitalSkola
Uncover The World of Digital Skills with Us

# Day 2: Intro to Machine Learning

## Understanding the concept of Machine Learning

# Table of Content

## What will We Learn Today?

1. Definisi *Machine Learning*
2. Jenis-jenis *Machine Learning (supervised and unsupervised learning)*
3. Konsep *train-test split*
4. *Overfitting & Underfitting*
5. *Case-scenario Machine Learning* di industri otomotif

# Mengapa ada *Machine Learning*?

Kamu punya data pelanggan dan ingin tahu: siapa yang mungkin beli motor listrik?

Dengan ML, komputer belajar dari data pembeli sebelumnya untuk memprediksi pembeli selanjutnya.

# Jenis-jenis Machine Learning

**Supervised machine learning:** belajar memprediksi nilai target dari data berlabel

- Classification (target values are discrete values)
- Regression (target values are continuous values)


**Classification**

**Regression**

# Jenis-jenis Machine Learning

**Unsupervised machine learning:** menemukan struktur dalam data yang tidak berlabel

- Find groups or similar instance in a data (clustering)
- Finding unusual pattern (outlier detection)

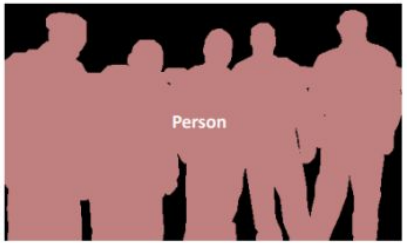



Image 1

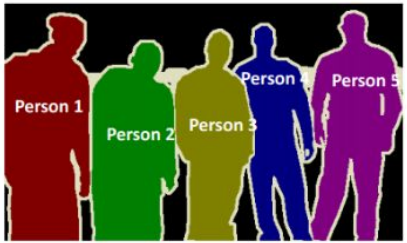
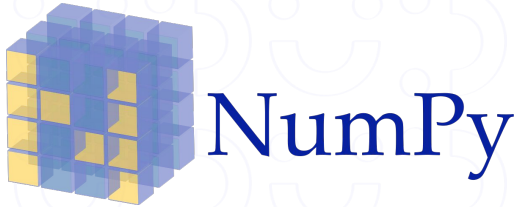



Image 2

# Tools Machine Learning

Beberapa package/library Python



# Konsep *Train-Test Split* & *Overfitting dan Underfitting*

# Train-Test Split

Untuk memastikan model tidak “ngafalin” data, kita harus uji kemampuannya pada data yang belum pernah dilihat.

## 📌 Proses:

- **Training set**: digunakan untuk *mengajari* model
- **Testing set**: digunakan untuk *mengukur kemampuan* model terhadap data baru

Biasanya: **80% untuk training, 20% untuk testing**

# Train-Test split

Contoh coding sederhana pada *train-test split*

python

Copy Edit

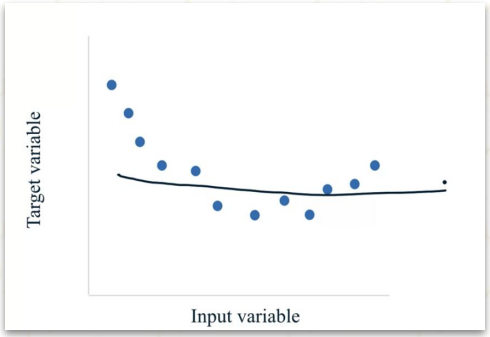
```python
# Memisahkan variabel independen (fitur) dan dependen (target)
X = data_encoded.drop(columns="is_promoted")  # Fitur yang digunakan untuk prediksi
y = data_encoded["is_promoted"]               # Target yang ingin diprediksi

# Melakukan pemisahan data latih dan data uji
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=43)
```

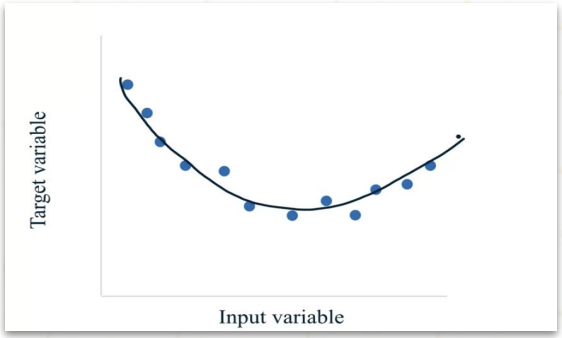
# Underfitting & Overfitting

DigitalSkola
Uncover The World Of Digital Skills With Us

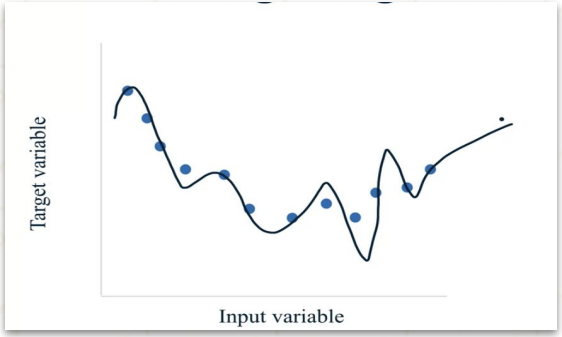
### Underfitting in Regression



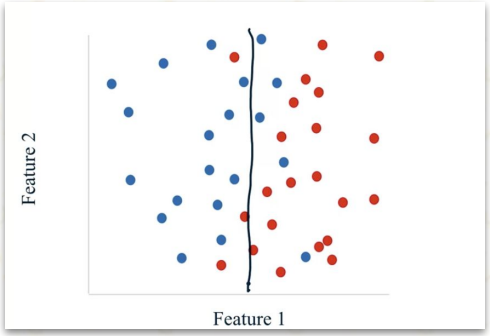

### Good Model in Regression



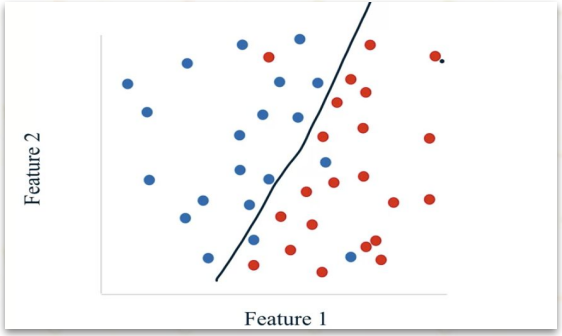

### Overfitting in Regression



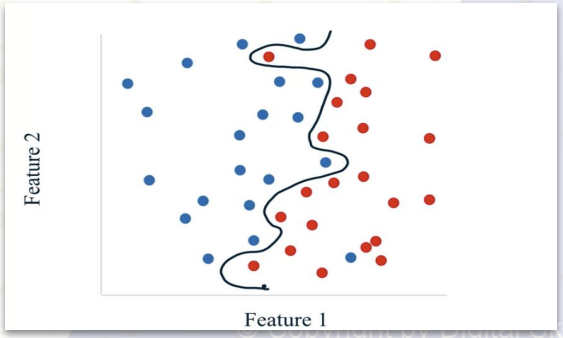

### Underfitting in Classification




### Good Model in Classification




### Overfitting in Classification



# Overfitting

Model **terlalu kompleks** dan **hafal data training** sampai ke detail, sehingga tidak bisa generalisasi ke data baru.

## 📌 Kenapa?

- Terjadi ketika model terlalu “ambisius”, ingin menyesuaikan semua titik data.
- Biasanya karena model terlalu “pintar” atau data terlalu sedikit.

# Underfitting

Model terlalu sederhana sehingga tidak bisa menangkap pola penting dalam data.

## 📌 Kenapa?

- Terjadi ketika model **tidak belajar cukup** dari data.
- Biasanya karena **fitur terlalu sedikit**

# Case-Scenario
## Machine Learning di Industri Otomotif

# Contoh ML Mode di Otomotif

| Use Case | Jenis ML | Deskripsi |
|---|---|---|
| **Deteksi kerusakan mobil secara otomatis dari sensor atau suara** | Supervised (klasifikasi) | Memprediksi apakah mobil butuh servis dari data mesin atau getaran yang direcord. |
| **Harga jual mobil bekas** | Supervised (regresi) | Prediksi harga berdasarkan tahun, jarak tempuh, jenis bahan bakar |
| **Perawatan prediktif (predictive maintenance)** | Supervised | Memprediksi kapan kendaraan akan rusak sebelum benar-benar rusak |
| **Segmentasi pelanggan bengkel atau dealer** | Unsupervised (clustering) | Mengelompokkan pelanggan berdasarkan frekuensi kunjungan atau jenis mobil |

DigitalSkola
Uncover The World of Digital Skills with Us

# Thank You.

DigitalSkola
Uncover The World of Digital Skills with Us

# Day 2: Classification Models

## Hands-on Building Machine Learning Classification Model

# Table of Content

## What will We Learn Today?

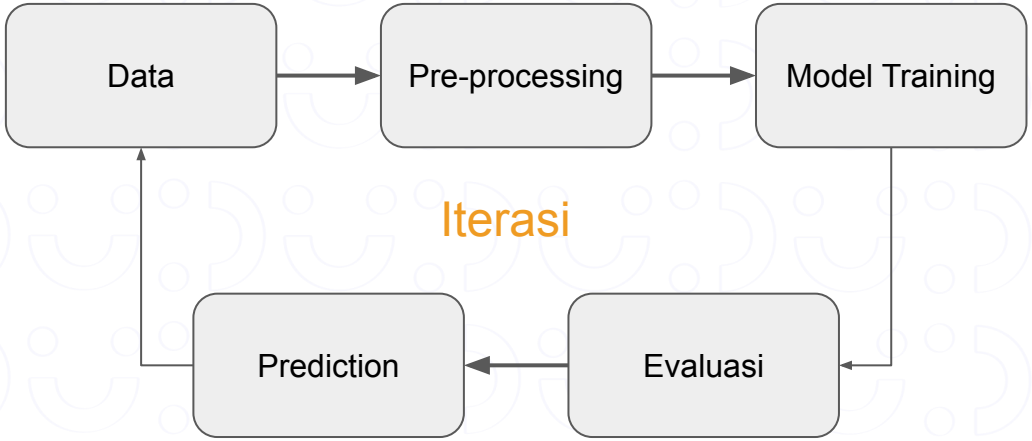
[1. Ilustrasi Machine Learning model untuk klasifikasi
2. Memahami beberapa jenis model
(Logistic Regression, Decision Tree, Random Forest)
[3. Hands-on Coding ML Model
4. Mengenal jenis *evaluation metrics*
*(Recall, Precision, F1 score)*

# Contoh Machine Learning

| No | Jenis *Real-Case* di Industri | Hasil output |
|---|---|---|
| 1 | Email spam detection | Spam, atau bukan Spam |
| 2 | Penerimaan Pinjaman | Diterima, atau ditolak |
| 3 | Diagnosis Penyakit | Positif, atau Negatif |
| 4 | Sistem Rekomendasi | Suka, atau tidak suka |

Hasil output dari analisis yang dilakukan, pola tersebut dapat dibuat menjadi model *machine learning* untuk memprediksi data baru yang masuk.

# Alur kerja Model ML Klasifikasi




# Bagaimana di AHM?

Apa yang menurut kamu di case-case di AHM yang memiliki kemungkinan untuk dilakukan projek *Machine Learning*?

# 2. Memahami Jenis-Jenis Klasifikasi ML Model

# Jenis-jenis Klasifikasi ML Model

Umumnya terdapat tiga jenis model dasar dalam klasifikasi *machine learning* model.

**Logistic Regression**

**Decision Tree**

**Random Forest**

# 4. Memahami Metrics Evaluasi Machine Learning Model

# Confusion Matrix

10, 000 PATIENTS

DIAGNOSIS

PATIENTS

| | Diagnosed Sick | Diagnosed Healthy |
|---|---|---|
| Sick | 1000 True Positives | 200 False Negatives |
| Healthy | 800 False Positives | 8000 True Negatives |

Two types of error:

1. **Type 1 error** (false positive), this is when we misdiagnose a healthy patient as sick
2. **Type 2 error** (false negative), this is when we misdiagnose a sick patient as healthy

# Accuracy

10, 000 PATIENTS

DIAGNOSIS

| PATIENTS | Diagnosed Sick | Diagnosed Healthy |
|---|---|---|
| Sick | 1000 True Positives | 200 False Negatives |
| Healthy | 800 False Positives | 8000 True Negatives |

*source: udacity machine learning nanodegree*

$$Accuracy = \frac{(TruePositive+TrueNegative)}{(TP+FP+FN+TN)}$$

$$= \frac{1.000 + 8.000}{10.000}$$

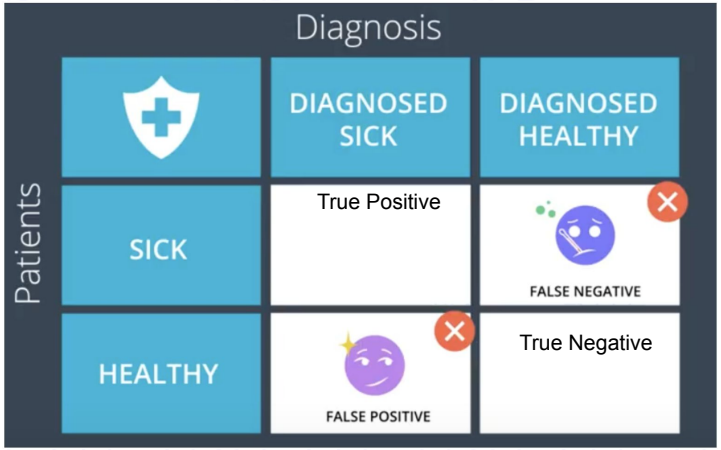
$$= 90\%$$

```
from sklearn.metrics import accuracy_score
accuracy_score (y_true, y_pred)
```

# Precision, Recall & F1 Score

False positive: when the patient is healthy, and you diagnose them as sick

False negative: when the patient is sick, and you diagnose them as healthy




*source: udacity machine learning nanodegree*

## Which one you choose?

$$Precision = \frac{(TruePositives)}{(TP+FP)}$$

*Out of all the points predicted to be positive, how many of them were actually positive?*

$$Recall = \frac{(TruePositive)}{(TP+FN)}$$

*Out of the points that are labeled positive, how many of them were correctly predicted as positive?*

F1 score: the average between precision and recall

#BertalentaDigital


# Hands-on Python

Buka Notebook Classification ML Model

DigitalSkola
Uncover The World of Digital Skills with Us

# Thank You.