DigitalSkola
Uncover The World of Digital Skills with Us

Day 1: Introduction & Refreshment Python for AI

End-to-end Process and Workflow

Table of Content

What will We Learn Today?

1. Set-up Environment Python for ML and AI (1 Hour)
2. Menggunakan Coding Assistant pada Google Colab
3. *Basic Coding* Python for ML and AI
4. Testing ML Model
5. Testing RAG

Overall Materials Python for AI

GitHub

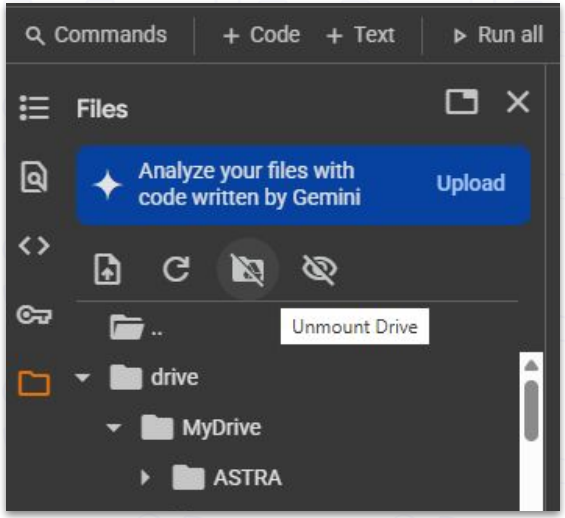
Download/Clone Material

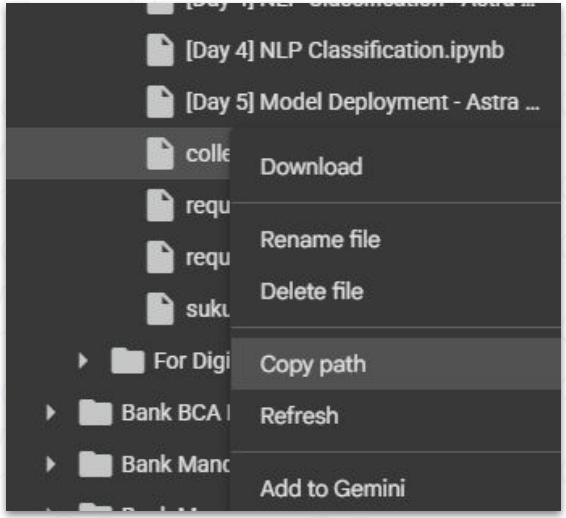
Python Environment in Google Colab

📌 Alur Set-up:

1. Google Colab Ready
2. Install package dari Requirements.txt
3. Membaca File dan Lokasi
4. Testing Running ML Model
5. Testing RAG Model

Lokasi File dan Membaca File





Copy path → Lokasi file yang nanti dibaca dan digunakan dalam membuat Machine Learning dan AI Model

#BertalentaDigital

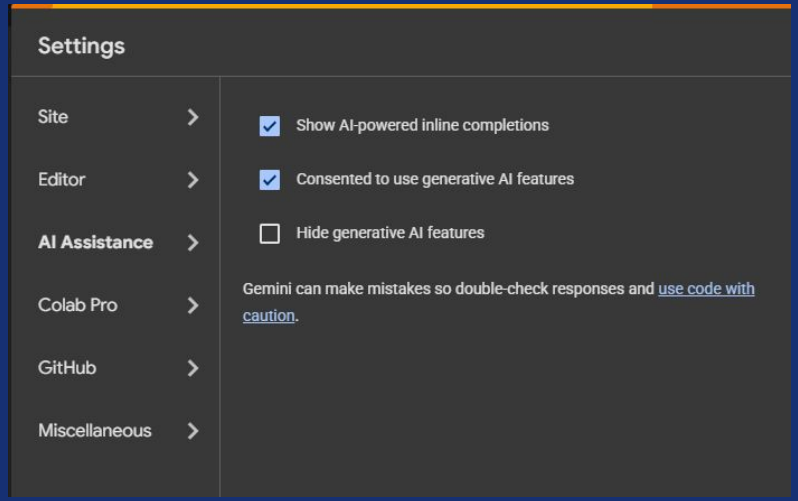
Coding Assistant Google Colab

#BertalentaDigital

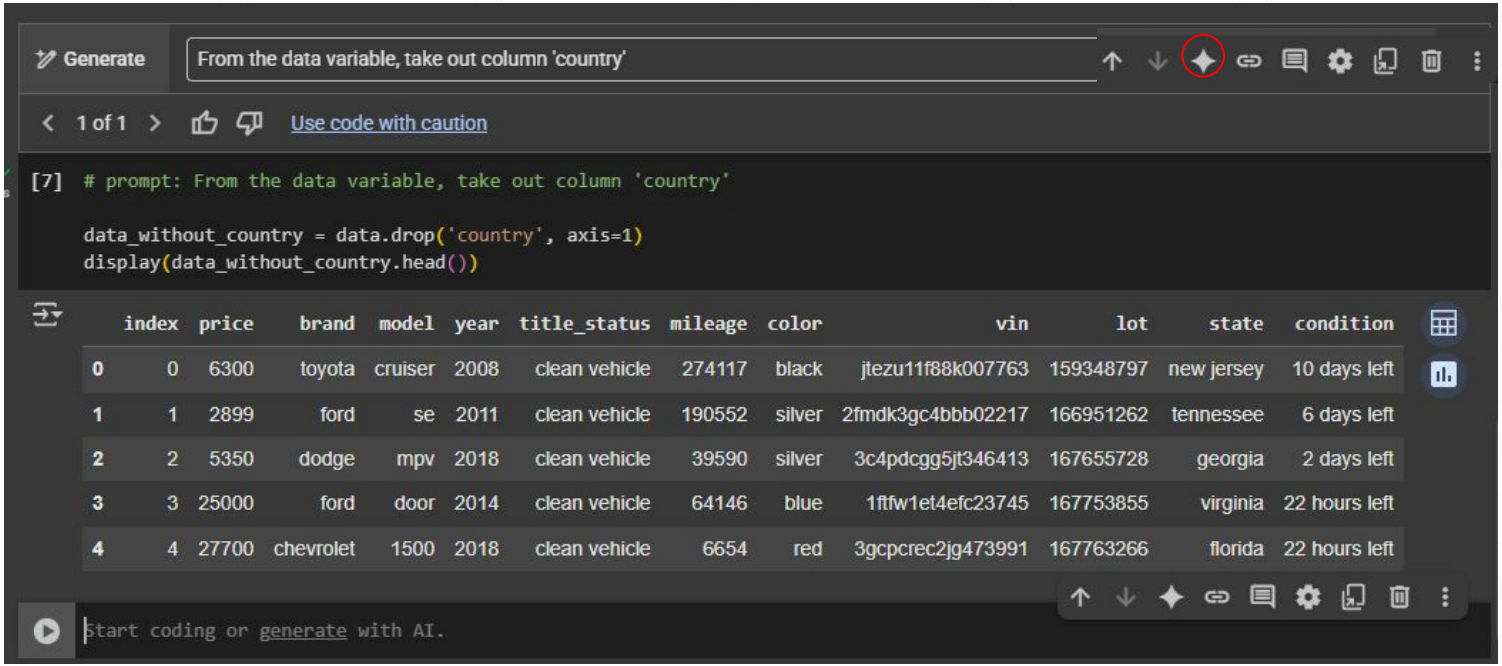
Coding Assistant in Google Colab

Mempermudah *code generation* selama *framework* sudah ditentukan.

Tools → Setting → AI Assistance

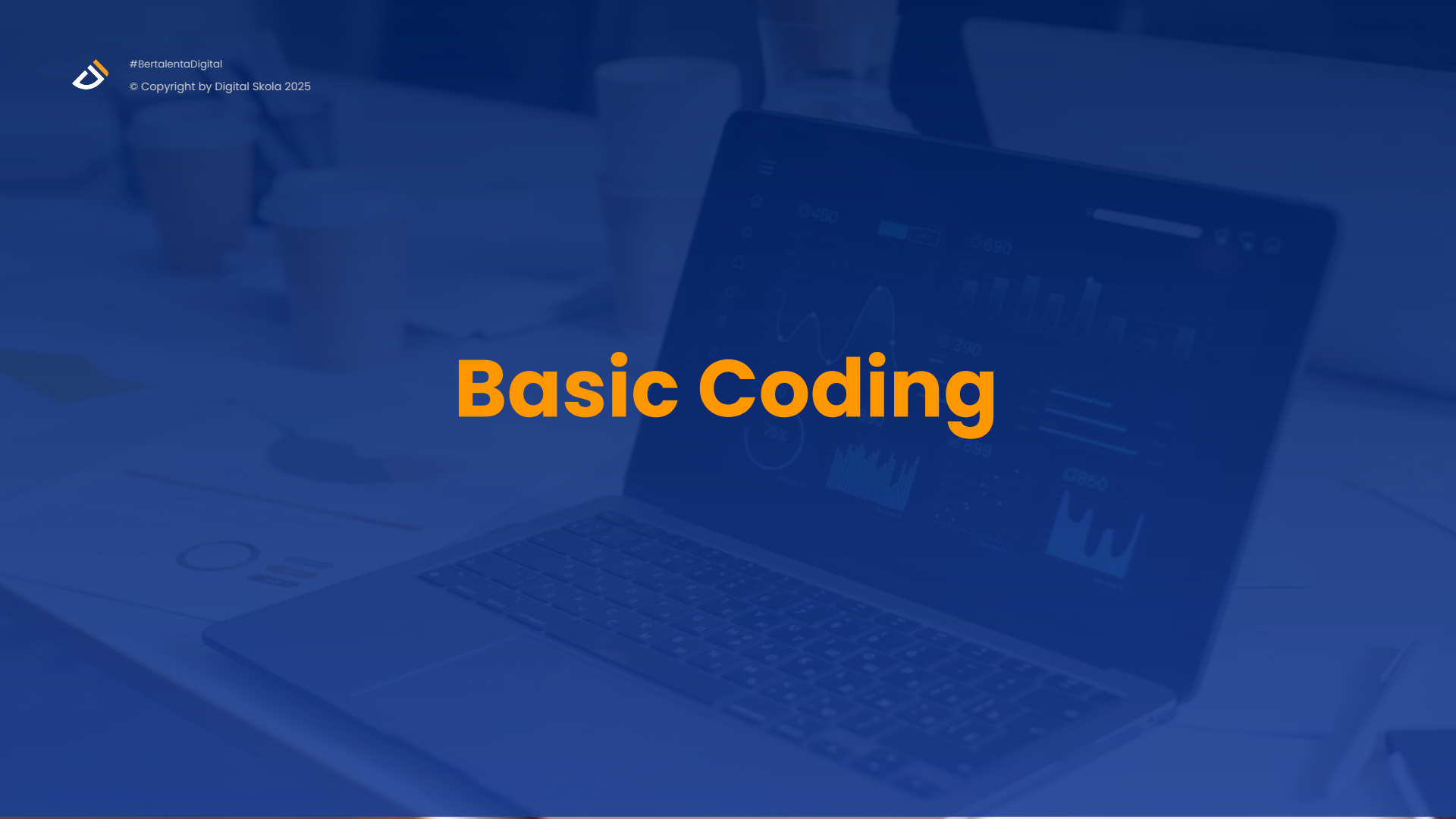


Cara Menggunakan Coding Assistant



	index	price	brand	model	year	title_status	mileage	color	vin	lot	state	condition
0	0	6300	toyota	cruiser	2008	clean vehicle	274117	black	jtezu11f88k007763	159348797	new jersey	10 days left
1	1	2899	ford	se	2011	clean vehicle	190552	silver	2fmdk3gc4bbb02217	166951262	tennessee	6 days left
2	2	5350	dodge	mpv	2018	clean vehicle	39590	silver	3c4pdcgg5jt346413	167655728	georgia	2 days left
3	3	25000	ford	door	2014	clean vehicle	64146	blue	1ftfw1et4efc23745	167753855	virginia	22 hours left
4	4	27700	chevrolet	1500	2018	clean vehicle	6654	red	3gcpcrec2jg473991	167763266	florida	22 hours left

#BertalentaDigital



Basic Coding

List Comprehension

```
clean_words = []

for w in word_tokens:
    if w not in stop_words:
        clean_words.append(w)
```

```
clean_words = [w for w in word_tokens if not w in stop_words]
```

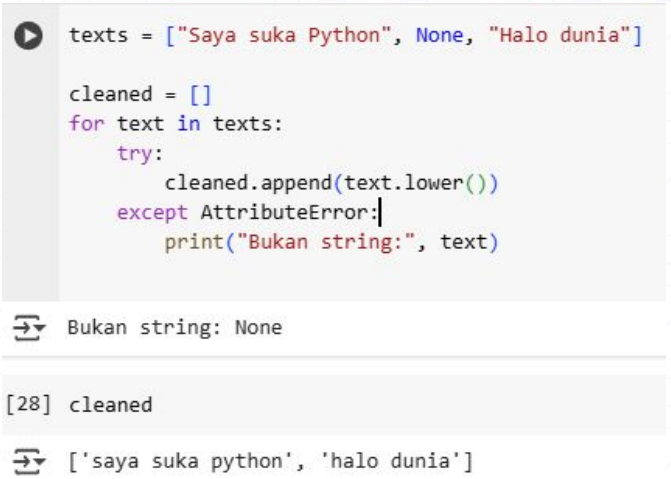
```
['Google+',
 'rolls',
 "'Stories",
 "'",
 'tricked',
 'photo',
 'playbackDov',
 'Charney',
 "'s",
 'Redeeming',
 'QualityWhite',
 'God',
 'adds',
 'Un',
```

Try and Except

try dan except digunakan untuk menangani error (exception) agar program tidak berhenti secara tiba-tiba saat terjadi kesalahan.

```
texts = ["Saya suka Python", None, "Halo dunia"]

cleaned = []
for text in texts:
    try:
        cleaned.append(text.lower())
    except AttributeError:
        print("Bukan string:", text)
```



Apply pada Function

```
def cek_status(nilai):
    if nilai <= 2010:
        return 'Brand Lama'
    else:
        return 'Brand Baru'

data['brand_status'] = data['year'].apply(cek_status)
```

	index	price	brand	model	year	title_status	mileage	color	state	country	condition	brand_status
0	0	6300	toyota	cruiser	2008	clean vehicle	274117	black	new jersey	usa	10 days left	Brand Lama
1	1	2899	ford	se	2011	clean vehicle	190552	silver	tennessee	usa	6 days left	Brand Baru
2	2	5350	dodge	mpv	2018	clean vehicle	39590	silver	georgia	usa	2 days left	Brand Baru
3	3	25000	ford	door	2014	clean vehicle	64146	blue	virginia	usa	22 hours left	Brand Baru
4	4	27700	chevrolet	1500	2018	clean vehicle	6654	red	florida	usa	22 hours left	Brand Baru

#BertalentaDigital

Testing

Machine Learning Model & RAG Model

DigitalSkola
Uncover The World of Digital Skills with Us

Thank You.

DigitalSkola
Uncover The World of Digital Skills with Us

Day 1: Data Manipulation

Bagian dari persiapan data dalam menggali informasi dari data

Table of Content

What will We Learn Today?

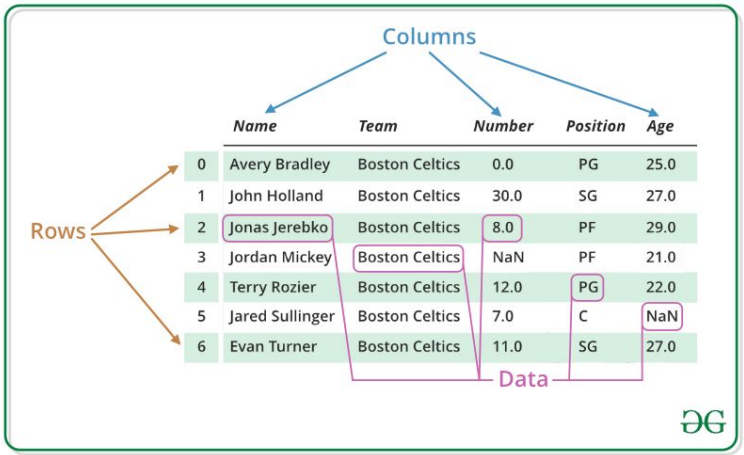
1. Membaca CSV/Excel File
2. *Filtering & Slicing*
3. Handling missing values
4. GroupBy dan agregasi Pivot Table

Mengapa perlu dilakukannya Data Manipulation?

- Menggabungkan Berbagai Sumber Data
- Mengelompokkan dan Menganalisis Pola
- Menyediakan Input yang Konsisten untuk Model ML

Membaca CSV/Excel File

Pandas Dataframe



Dataframe adalah struktur data yang berlabel 2 dimensi dengan kolom yang memiliki tipe data yang berbeda. Seperti halnya yang ada di Excel.

```
import pandas as pd

# CSV File
df = pd.read_csv('vehicle_data.csv')
```

```
# Excel File
df = pd.read_excel('vehicle_data.xlsx', sheet_name='overview')
```

Filtering & Slicing

📌 Konsep Utama:

- Index Kolom & Baris → Mengambil kolom dan baris yang diinginkan untuk preparation machine learning.
- Data Point → Umumnya, untuk persiapan menganalisis pola dari data

Slicing Column

```
[8] # Ambil kolom yang diinginkan
    data[['price','brand','model']]
```

	price	brand	model
0	6300	toyota	cruiser
1	2899	ford	se
2	5350	dodge	mpv
3	25000	ford	door
4	27700	chevrolet	1500
...	...	...	...
2494	7800	nissan	versa
2495	9200	nissan	versa
2496	9200	nissan	versa
2497	9200	nissan	versa
2498	9200	nissan	versa

```
# Hilangkan kolom yang tidak diinginkan
data.drop(columns=['price'])
```

	index	brand	model	year	title_status	mileage	color	vin	lot	state	country	condition
0	0	toyota	cruiser	2008	clean vehicle	274117	black	jtezu11f88k007763	159348797	new jersey	usa	10 days left
1	1	ford	se	2011	clean vehicle	190552	silver	2fmdk3gc4bbb02217	166951262	tennessee	usa	6 days left
2	2	dodge	mpv	2018	clean vehicle	39590	silver	3c4pdcgg5jt346413	167655728	georgia	usa	2 days left
3	3	ford	door	2014	clean vehicle	64146	blue	1ftfw1et4efc23745	167753855	virginia	usa	22 hours left
4	4	chevrolet	1500	2018	clean vehicle	6654	red	3gcpcrec2jg473991	167763266	florida	usa	22 hours left
...	...	...	...	...	...	...	...	...	...	...	...	...
2494	2494	nissan	versa	2019	clean vehicle	23609	red	3n1cn7ap9kl880319	167722715	california	usa	1 days left
2495	2495	nissan	versa	2018	clean vehicle	34553	silver	3n1cn7ap5jl884088	167762225	florida	usa	21 hours left
2496	2496	nissan	versa	2018	clean vehicle	31594	silver	3n1cn7ap9jl884191	167762226	florida	usa	21 hours left
2497	2497	nissan	versa	2018	clean vehicle	32557	black	3n1cn7ap3jl883263	167762227	florida	usa	2 days left
2498	2498	nissan	versa	2018	clean vehicle	31371	silver	3n1cn7ap4jl884311	167762228	florida	usa	21 hours left

Filtering Data Point

```
# Filter data
# tahun 2016 - 2018
# warna merah

data[(data['year'] >= 2016) & (data['year'] <= 2018) & (data['color'] == 'red')]
```

	index	price	brand	model	year	title_status	mileage	color	vin	lot	state	country	condition
4	4	27700	chevrolet	1500	2018	clean vehicle	6654	red	3gcpcrec2jg473991	167763266	florida	usa	22 hours left
17	17	16500	buick	encore	2018	clean vehicle	20002	red	kl4cj1sb6jb688844	167763275	tennessee	usa	22 hours left
42	42	30500	chevrolet	1500	2018	clean vehicle	30442	red	3gcukrec4jg204915	167763393	tennessee	usa	22 hours left
43	43	6330	ford	mpv	2017	clean vehicle	38123	red	2fmpk3j85hbb79415	167656129	texas	usa	2 days left
50	50	12710	gmc	mpv	2017	clean vehicle	39886	red	1gks2bkc7hr133811	167692513	california	usa	20 hours left
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2300	2300	7500	nissan	versa	2018	clean vehicle	65184	red	3n1cn7ap3jl833592	167724491	california	usa	2 days left
2334	2334	12900	nissan	door	2016	clean vehicle	98354	red	1n6ad0ev8gn783236	167612818	new jersey	usa	2 hours left
2373	2373	15000	nissan	rogue	2017	clean vehicle	28918	red	jn8at2mv7hw264732	167753299	virginia	usa	21 hours left
2392	2392	11800	nissan	door	2017	clean vehicle	60158	red	1n4al3ap1hc129888	167754120	pennsylvania	usa	3 days left
2417	2417	375	nissan	door	2017	salvage insurance	1	red	3n1ce2cp0hl360793	167619908	north carolina	usa	2 days left

Handling Missing Value

Handling Missing Value

"Menghapus atau mengisi data yang hilang (NaN), karena algoritma ML tidak bisa memproses missing value secara langsung"

	Columns Type	Amount of Null Values	Percentage of Null Values
Unnamed: 0	int64	0	0.0
Loan_ID	object	0	0.0
Gender	object	10	2.04
Married	object	1	0.2
Dependents	object	9	1.83
Education	object	0	0.0
Self_Employed	object	29	5.91
ApplicantIncome	int64	0	0.0
CoapplicantIncome	float64	0	0.0
LoanAmount	float64	16	3.26
Loan_Amount_Term	float64	13	2.65
Credit_History	float64	43	8.76
Property_Area	object	0	0.0
Loan_Status	int64	0	0.0

Simple Imputer

Simple Imputer digunakan untuk **mengisi (imputasi) nilai yang hilang (missing values)** pada dataset.

🎯 Mengapa Penting?

Dalam machine learning, nilai hilang (NaN) harus ditangani karena:

- Sebagian besar algoritma **tidak bisa bekerja dengan data yang memiliki missing values**
- Imputasi membantu mempertahankan jumlah data tanpa membuang baris (row)

Strategi	Penjelasan
'mean'	Mengganti nilai NaN dengan **rata-rata kolom**
'median'	Mengganti dengan **nilai tengah** kolom
'most_frequent'	Mengganti dengan **nilai yang paling sering muncul**
'constant'	Mengganti dengan **nilai tetap (misal: 0 atau "missing")**

Pivot Table dan Agregasi

Pivot Table

Pivot table bertujuan untuk memberikan informasi berupa **agregasi suatu data** dengan melampirkan isi data pada nama kolom tertentu

'mean'	menghitung nilai rata-rata
'sum'	menghitung jumlah nilai data
'count'	menghitung jumlah baris dari *non-null* data
'min'	menghitung nilai minimum
'max'	menghitung nilai maximum
'median'	menghitung nilai tengah
pd.Series.nunique:	menghitung jumlah unique dari *non-null* data

Contoh Pivot Table

🎯 Baris dan Agregasi

```
# Pivot data untuk baris dan nilai yang diagregasi
pd.pivot_table(data, index='model', values='price',
               aggfunc='median').reset_index()
```

	model	price
0	1500	27700.0
1	2500	31951.0
2	2500hd	32500.0
3	300	19350.0
4	3500	45050.0

🎯 Baris, Kolom, dan Agregasi

```
# Pivot data baris, kolom, dan nilai yang diagregasi
cek = pd.pivot_table(data, index='model', columns='title_status',
                     values='price', aggfunc='mean').reset_index()

# Menampilkan baris, kolom, dan nilai agregasi
cek.columns.name = None
cek
```

	model	clean vehicle	salvage insurance
0	1500	27265.128205	NaN
1	2500	31612.750000	NaN
2	2500hd	32500.000000	NaN
3	300	19100.000000	NaN
4	3500	38325.000000	NaN

#BertalentaDigital

Hands-on Python

Buka Notebook Classification ML Model

DigitalSkola
Uncover The World of Digital Skills with Us

Thank You.

DigitalSkola
Uncover The World of Digital Skills with Us

Day 1: Data Cleaning & Pre-Processing

Membersihkan data dan memproses data untuk digunakan dalam Machine Learning

Table of Content

What will We Learn Today?

1. Mengubah tipe data
2. Deteksi Outlier dan Penanganan
3. *Encoding Categorical Data*
4. Scaling & Normalization

Mengubah Tipe Data

Tidak semua tipe data dari hasil *data extraction* bersih dan siap untuk dianalisis.

🚨 Contoh

```
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

Case Mengubah Tipe Data

🗓️1. String → DateTime

Tujuan: Tanggal atau waktu sering terbaca sebagai string padahal perlu digunakan sebagai tipe datetime.

```
# Mengganti tipe data string menjadi tanggal
dataset['Order Date'] = pd.to_datetime(dataset['Order Date'])
```

🗓️ 2. Float → Integer

Tujuan: Menyesuaikan dengan kebutuhan analisis atau agar hasil tidak muncul angka desimal jika tidak diperlukan.

```
# Mengganti tipe data float menjadi integer
dataset['Profit'] = dataset['Profit'].astype(int)
```

Mendeteksi Outlier

Mengapa Outlier perlu Dideteksi?

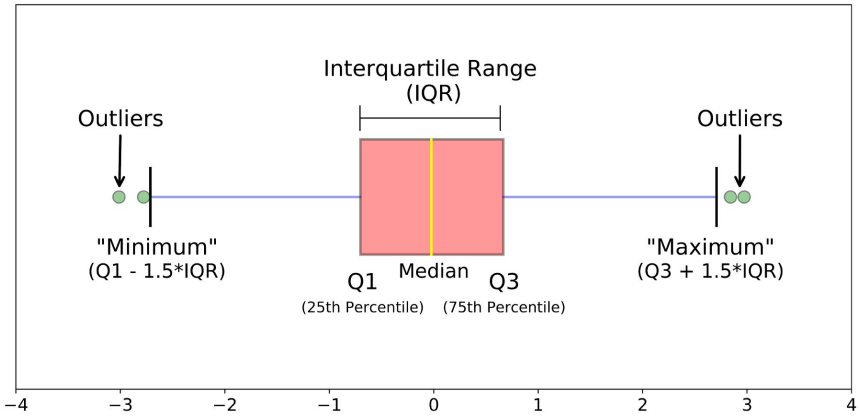
- Memberikan informasi yang keliru dalam analisis.
- Machine learning model jadi tidak akurat.
- Decision-making jadi tidak tepat sasaran.

Contoh Data Outlier

Merek	Jenis Sparepart	Harga
A	Aki Motor	350.000
B	Aki Motor	280.000
C	Aki Motor	500.000
D	Aki Motor	380.000
E	Aki Motor	5.000.000

Rata-rata seharusnya Aki Motor di kisaran 200.000 - 500.000
Namun dari data di atas, rata-rata Aki Motor adalah 1.300.000

Handle Outlier



Maximum: nilai pengamatan terbesar

Minimum: nilai pengamatan terkecil

First Quartile: nilai tengah antara **minimum** dan **median**. Penjelasannya 25% data terendah ada di Q1

Second Quartile (Median): nilai tengah antara titik Q1 dan Q3.

Third Quartile: nilai tengah antara **maximum** dan **median**. Penjelasannya 75% data tertinggi ada di Q3

Encoding Categorical Data

Encoding Categorical Data

Perlu dilakukan karena

1. Machine learning hanya dapat membaca data berupa data numerik
2. Model Membutuhkan Representasi Numerik yang Konsisten
3. Menghindari Ambiguitas dan Kesalahan Interpretasi

Label Encoding

Label encoding mengubah nilai kategori menjadi angka integer.

🎯 Misalnya

jika suatu fitur memiliki tiga kategori, maka masing-masing diberi nilai 0, 1, dan 2.

```
from sklearn.preprocessing import LabelEncoder

df['fuel_type'] = ['Petrol', 'Diesel', 'Electric', 'Diesel']
encoder = LabelEncoder()
df['fuel_type_encoded'] = encoder.fit_transform(df['fuel_type'])
```

fuel_type	fuel_type_encoded
Petrol	2
Diesel	0
Electric	1
Diesel	0

One-Hot Encoding

Mengubah kategori menjadi beberapa kolom biner (0/1). Setiap kolom mewakili satu kategori.

```
import pandas as pd

df = pd.DataFrame({'fuel_type': ['Petrol', 'Diesel', 'Electric', 'Diesel']})
df_encoded = pd.get_dummies(df, columns=['fuel_type'], prefix='fuel')
```

fuel_type_Diesel	fuel_type_Electric	fuel_type_Petrol
0	0	1
1	0	0
0	1	0
1	0	0

Kapan Memakai

Label Encoding & One-Hot Encoding

Kondisi Dataset	Disarankan
Kategori sedikit dan tidak ada makna urutan	One-Hot
Kategori sangat banyak	Label
Model berbasis pohon keputusan (tree-based)	Label
Model berbasis jarak atau regresi/logistik	One-Hot
Ada makna urutan antar kategori	Label (atau gunakan ordinal encoder)

Scaling & Normalization

Scaling & Normalization

Perlu dilakukan karena

Menyesuaikan skala data agar semua fitur memiliki kontribusi yang seimbang, terutama penting untuk algoritma berbasis jarak seperti KNN, SVM, atau Gradient Descent.

🛑 **Kapan Tidak Perlu Scaling?**

- Model berbasis pohon: **Decision Tree, Random Forest, XGBoost** tidak memerlukan scaling.
- Fitur kategorikal (sudah di-encode): scaling tidak relevan.

⚠️ **Tips Penting:**

- Scaling harus dilakukan **setelah** data split (fit scaler pada **train set**, transform ke **test set**).

Perbedaan Scaling & Normalization

Scaling

Menyesuaikan fitur agar memiliki mean = 0 dan std = 1

$$y = \frac{x - min(x)}{max(x) - \min(x)}$$

Normalization

Mengubah rentang fitur ke 0–1 atau -1–1

$$x' = \frac{x - \mu}{\max(x) - \min(x)}$$

Perbedaan Scaling & Normalization

engine_size	Scaling (standardization)	Normalization
1500	-1.18	0.0
2500	-0.11	0.5
3000	0.46	0.67
4500	0.83	1

#BertalentaDigital

Hands-on Python

Buka Notebook Classification ML Model

DigitalSkola
Uncover The World of Digital Skills with Us

Thank You.