Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering & Data Preprocessing

Muhammad Karov Ardava Barus
Machine Learning Mentor @GDGoC Tel-U,
Data Science Student | Telkom University

Ardavaa ardava-barus @rdavaa_

Hello World, I'm Ardava

Big Data Research Assistant

Focuses on Large Language Model Research and Machine Learning related topic.
Oct 2024 - Present

Head of Human Resource Development Department (PSDM)

Jan 2024 - Present

Undergraduate Data Science Student

2nd Year Undergraduate Data Science Student.

Presentation Link

ristek.link/GDGOC-ML-SG4

Google Developer Groups
On Campus • Telkom University Bandung

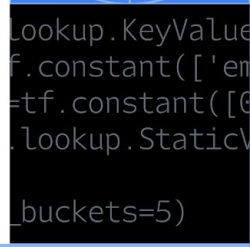
Today's Topic

Data Preprocessing

- Definition
- Handling Missing & Duplicated Data
- Outliers Engineering
- Feature Scaling

Feature Engineering

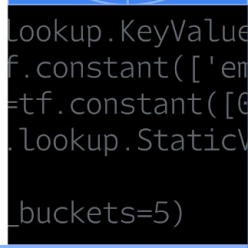
- Definition
- Categorical Encoding
- Rare Label
- Variable Transformation
- Date Time Feature



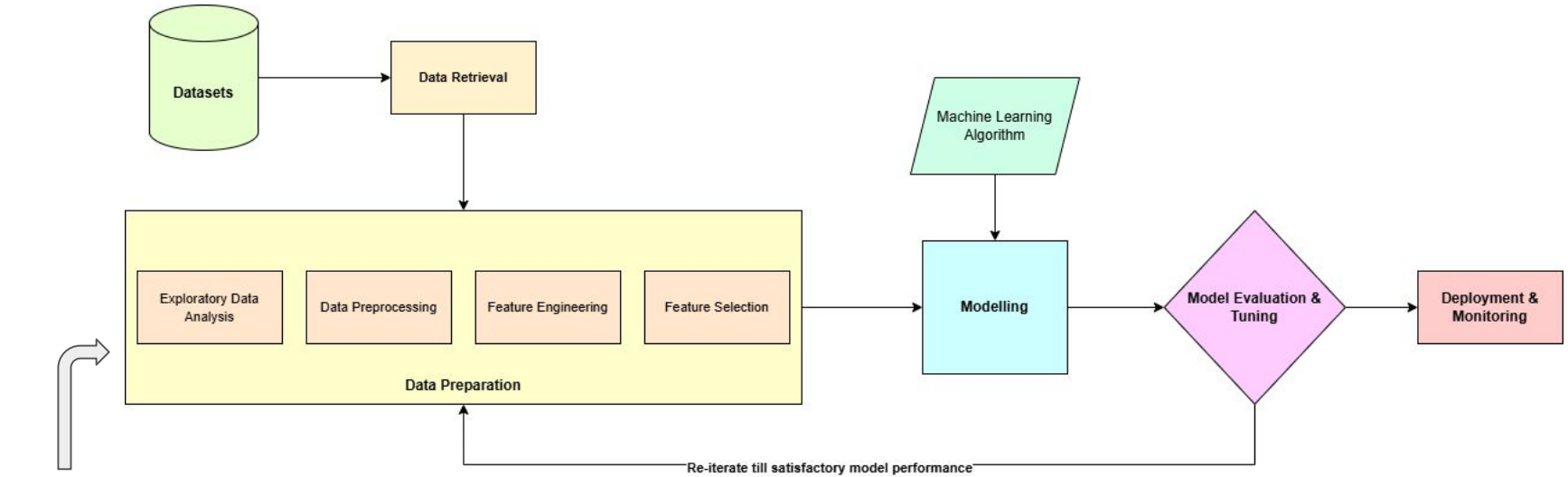
Google Developer Groups
On Campus • Telkom University Bandung

Data Preprocessing

Definition



Machine Learning Workflow



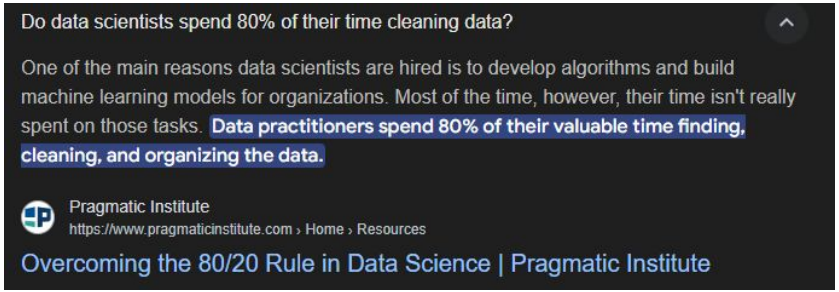
Data Preprocessing: Definition

Apa itu Data Preprocessing?

- **Data preprocessing** adalah teknik yang digunakan untuk mengubah data mentah dalam format yang berguna dan efisien.
- Proses ini dilakukan untuk **memperbaiki** kekurangan data mentah, sehingga algoritma machine learning lebih mudah untuk memahaminya.

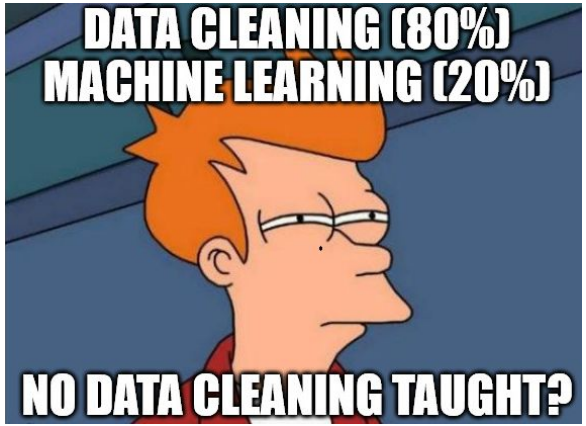
Data Preprocessing: Definition

Kenapa butuh Data Preprocessing?



Jawabannya:

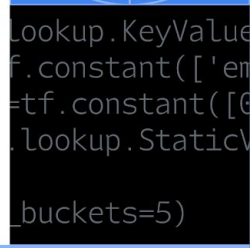
Kualitas data tidak ada, kualitas hasil prediksi juga tidak ada!



Google Developer Groups
On Campus • Telkom University Bandung

Data Preprocessing

Handling Missing & Duplicated Data



Data Preprocessing:
Handling Missing & Duplicated Data

Missing Data

- **Data yang hilang**, atau **nilai yang hilang**, terjadi ketika tidak ada data yang disimpan untuk suatu observasi tertentu dalam sebuah variabel.
- **Data yang hilang** adalah kejadian umum di sebagian besar kumpulan data.
- **Data yang hilang** dapat memiliki dampak yang signifikan terhadap kesimpulan yang dapat diambil dari data tersebut.

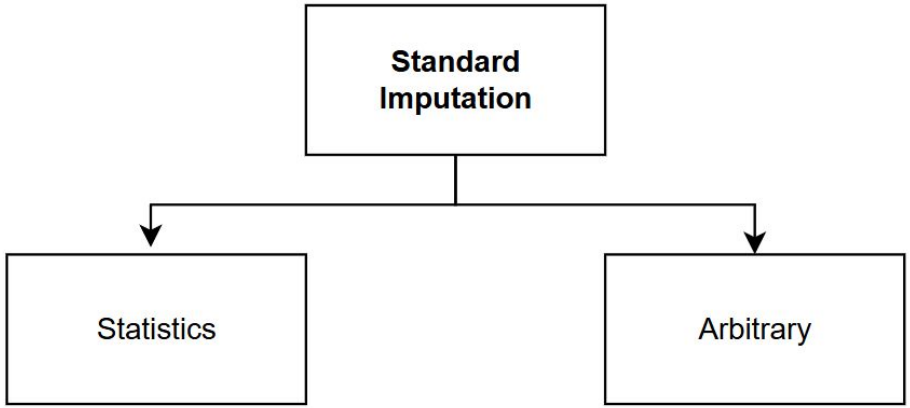
Sebuah nilai hilang karena **lupa, hilang**, atau **tidak disimpan dengan benar**.

Contoh: sebuah variabel dibuat dari pembagian dua variabel, dan penyebutnya bernilai 0.

Contoh: ketika mencocokkan data dengan kode pos atau tanggal lahir untuk memperkaya dengan lebih banyak variabel, tetapi kode pos atau tanggal lahir salah atau tidak ada, maka variabel baru akan bernilai NA.

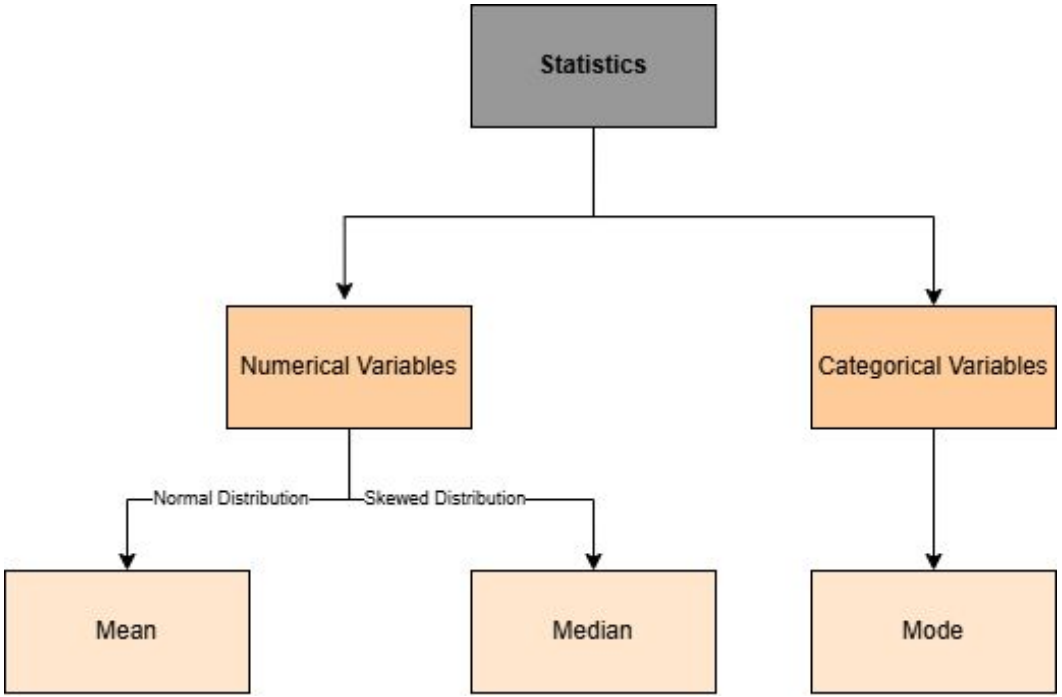
Data Preprocessing:
Handling Missing & Duplicated Data

Missing Data: Standard Imputation

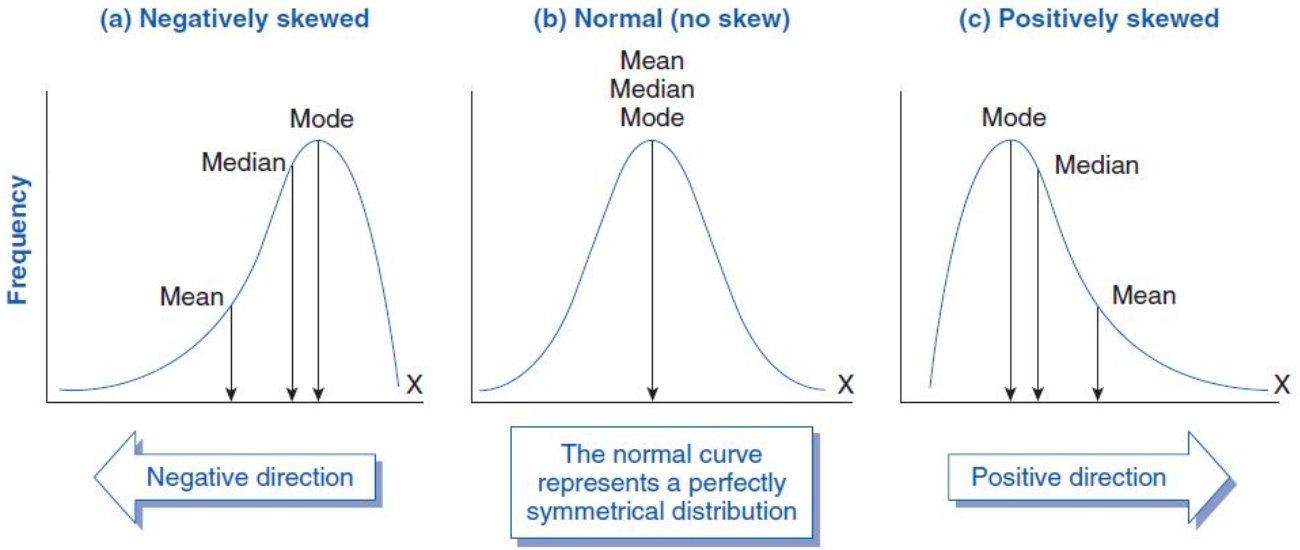


Data Preprocessing:
Handling Missing & Duplicated Data

Missing Data: Statistics Based



Data Preprocessing:
Handling Missing & Duplicated Data



https://www.biologyforlife.com/skew.html

Data Preprocessing:
Handling Missing & Duplicated Data

Price
100
90
50
40
20
100
60
120
200

Mean = 86.66

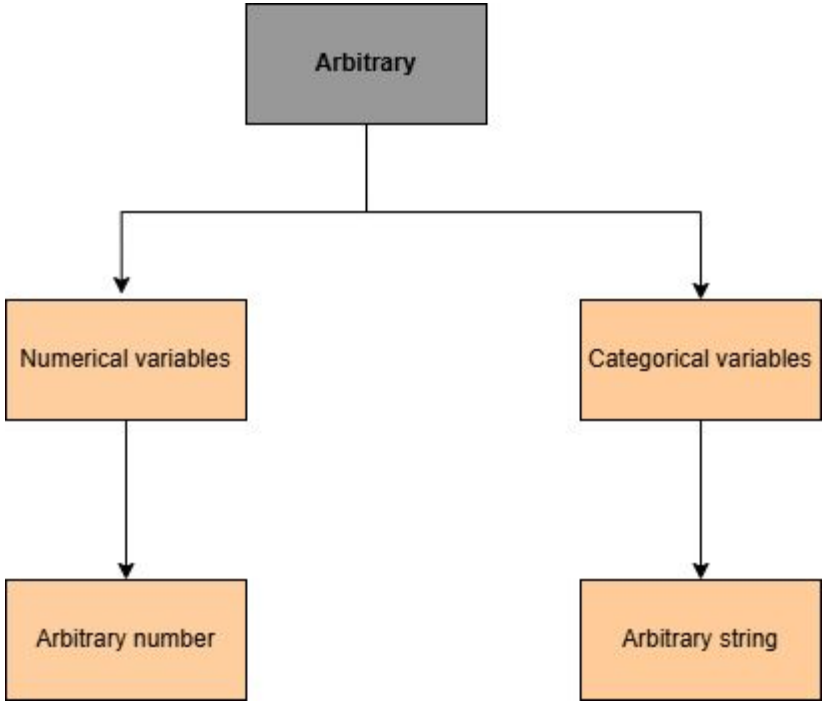
Median = 90

Price
100
90
50
40
20
100
86.66
60
120
86.66
200

Data Preprocessing:
Handling Missing & Duplicated Data

Arbitrary Values

- Imputasi **nilai arbitrer** terdiri dari menggantikan semua kejadian nilai yang hilang (NA) dalam sebuah variabel dengan nilai arbitrer tertentu.
- **Nilai arbitrer** yang biasanya digunakan adalah 0, 999, -999 (atau kombinasi angka 9 lainnya), atau -1 (jika distribusi bernilai positif).
- Cocok untuk variabel numerik dan kategorikal.
    - Untuk variabel kategorikal → gunakan label “Missing”.



Data Preprocessing:
Handling Missing & Duplicated Data

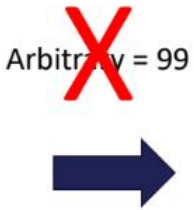
Price
100
90
50
40
20
100
60
120
200

Arbitrary = 999

Price
100
90
50
40
20
100
999
60
120
999
200

Data Preprocessing:
Handling Missing & Duplicated Data

Price
100
90
50
40
20
100
60
120
200



Price
100
90
50
40
20
100
999
60
120
999
200

Kita seharusnya meng-imputasi nilai arbitrary yang sangat jauh dari nilai distribusi datanya.

Missing Data

Check Missing Values

```python
import pandas as pd

df = pd.read_csv('iris.csv')

# Check for missing values
df.isna().sum()
```

✓ 0.0s

```
sepal length (cm)      0
sepal width (cm)       0
petal length (cm)      0
petal width (cm)       0
species                0
sepal_length         150
petal_width          150
dtype: int64
```

Fill With Mean

```python
# fill with mean
df['petal length (cm)'].fillna(df['petal length (cm)'].mean(), inplace=True)
```

Fill With Median

```python
# fill with median
df['sepal_length'] = df['sepal_length'].fillna(df['sepal_length'].median())
```

Fill With Mode

```python
# fill with mode
df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
```

Missing Data

Fill with numerical arbitrary value

```
# fill with arbitrary value (numerical)
df.fillna(-1, inplace=True)
```

Fill with categorical arbitrary value

```
# fill with arbitrary value (categorical)
df.fillna('missing', inplace=True)
```

Data Preprocessing:
Handling Missing & Duplicated Data

Missing Data: Row Deletion (Complete Case Analysis)

Gender	Price	Make	Engine
Female	100	Ford	2000
	90	Ford	2000
Male	50	Kia	1500
Male	60	Kia	
Female	120	Nissan	3000
Female		BMW	4500
Male	200	BMW	4500

Gender	Price	Make	Engine
Female	100	Ford	2000
Male	50	Kia	1500
Female	120	Nissan	3000
Male	200	BMW	4500

Observations with missing values are removed

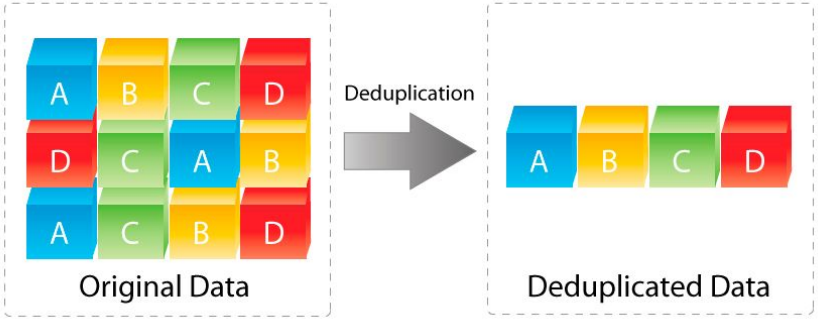
Data Preprocessing:
Handling Missing & Duplicated Data

Duplicated Data

- **Data duplikat,** atau **entri duplikat,** terjadi ketika satu atau lebih baris data yang sama muncul lebih dari sekali dalam suatu kumpulan data.
- **Data duplikat** adalah masalah umum dalam pengolahan data, terutama saat data dikumpulkan dari berbagai sumber.
- **Data duplikat** dapat menyebabkan analisis yang tidak akurat, meningkatkan ukuran dataset secara tidak perlu, dan memperlambat proses pengolahan data.

Lalu bagaimana cara meng-*handle* nya?

Simple, Dengan cara **menghapus data duplikat** tersebut.



Ilustrasi

https://www.hsb.nl/the-importance-of-deduplication-and-adjudication-in-identity-management-solutions/

Duplicated Data

1

```
# contoh data yang duplikat
data = {
    'name': ['A', 'B', 'C', 'A', 'B', 'C'],
    'salary': [100, 200, 300, 100, 200, 300]
}

df = pd.DataFrame(data)

# melihat baris data yang duplikat
df.duplicated()
```

✓ 0.0s

```
0    False
1    False
2    False
3     True
4     True
5     True
dtype: bool
```

```
# melihat jumlah baris data yang duplikat
df.duplicated().sum()
```

✓ 0.0s

```
3
```

Preview tabel

	name	salary
0	A	100
1	B	200
2	C	300
3	A	100
4	B	200
5	C	300

2

Menghapus data duplikat

```
df.drop_duplicates(inplace=True)

df
```

✓ 0.0s

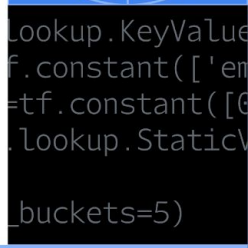
Preview tabel setelah penghapusan data duplikat

	name	salary
0	A	100
1	B	200
2	C	300

Google Developer Groups
On Campus • Telkom University Bandung

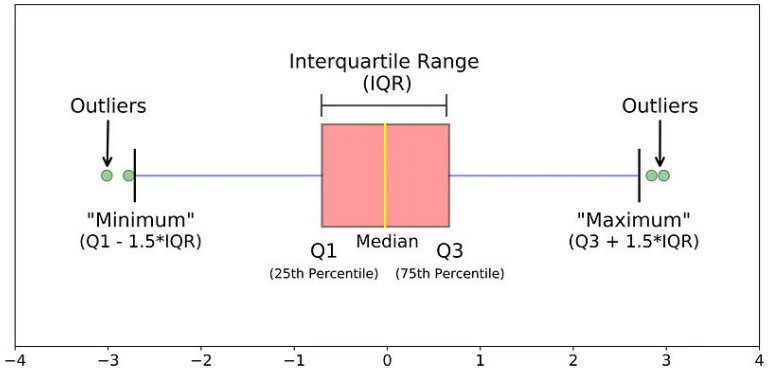
Data Preprocessing

Outliers Engineering



Data Preprocessing:
Outliers Engineering

Apa itu Outlier?



- **Outlier** adalah data yang sangat berbeda atau menyimpang jauh dari data lainnya dalam suatu kumpulan data.
- **Penyebab**:
  - Kesalahan data: Kesalahan saat pengumpulan atau penginputan data.
  - Data ekstrem yang valid: Data yang memang benar-benar berbeda dari yang lain, misalnya pendapatan seorang CEO dibandingkan dengan karyawan biasa dalam satu perusahaan.
  - Anomali: Peristiwa yang tidak biasa atau jarang terjadi.

https://medium.com/@agarwal.vishal819/outlier-detection-with-boxplots-1b6757fafa21

Data Preprocessing: Outliers Engineering

Let's take an example to check what happens to a dataset with and without outliers

	Data without outlier	Data with outlier
Data	1,2,3,3,4,5,4	1,2,3,3,4,5,**400**
Mean	3.142	**59.714**
Median	3	3
Standard Deviation	1.345185	**150.057**

Data dengan outlier memiliki rata-rata dan standar deviasi yang berbeda secara signifikan. Pada skenario pertama, kita akan mengatakan bahwa rata-rata adalah 3,14. Namun dengan adanya pencilan, rata-rata melonjak menjadi 59,71. Hal ini akan mengubah estimasi sepenuhnya.

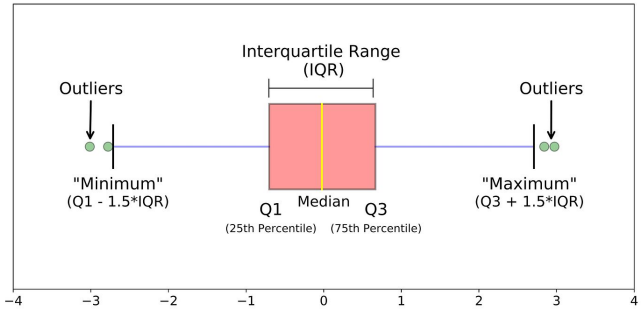
Data Preprocessing:
Outliers Engineering

The above meme makes you better understanding of outlier.

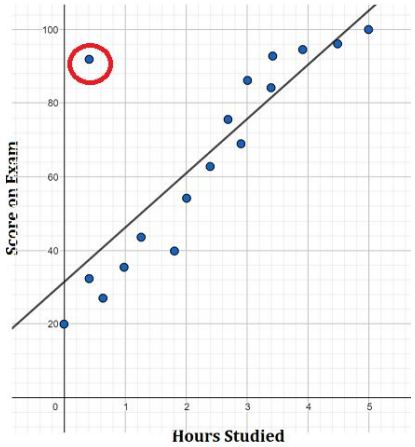
Data Preprocessing: Outliers Engineering

How to Visualize Outliers?

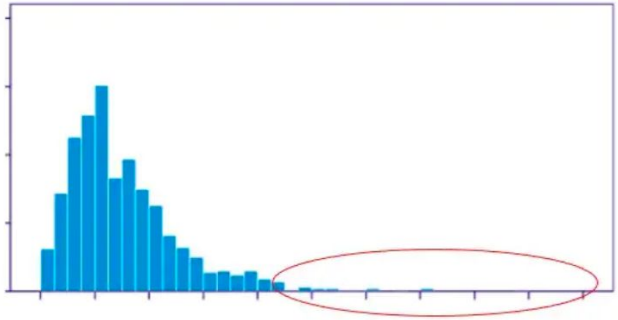
Boxplot



Scatter Plot



Histogram Plot



Data Preprocessing: Outliers Engineering

How to Handle Outliers?

- **Remove the outlier**
  - **When**: Jika jelas-jelas merupakan error (e.g., a typo, incorrect measurement).
  - **Contoh**: Dalam kumpulan data usia siswa, Anda menemukan seorang siswa yang terdaftar berusia 200 tahun. Ini jelas merupakan kesalahan dan harus dihapus..

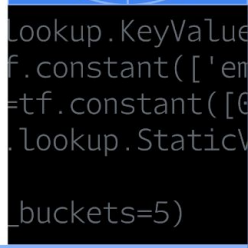
- **Transform the data**
  - **When**: Outlier itu nyata tetapi secara signifikan mempengaruhi analisis.
  - **Methods**:
    - **Log transformation**: Mengurangi dampak dari nilai yang sangat besar.
    - **Winsorization**: Mengganti nilai ekstrim dengan nilai yang tidak terlalu ekstrem (misalnya persentil ke-5 dan ke-95).
    - **Standardization (z-score)**: Mentransformasi data agar memiliki rata-rata 0 dan deviasi standar 1, yang dapat membantu mengurangi pengaruh outlier
  - Contoh: Dalam kumpulan data harga rumah, beberapa rumah yang sangat mahal dapat membuat hasil yang tidak sesuai. Transformasi log harga dapat membuat data terdistribusi secara lebih normal dan lebih mudah dianalisis.

- **Keep it**
  - **When**: Outlier adalah nyata dan penting untuk analisis.
  - **Contoh**: Dalam kumpulan data ukuran hewan, paus yang sangat besar akan menjadi pencilan. Namun, hal ini sangat penting untuk memahami keragaman ukuran hewan dan harus disertakan dalam analisis.

Google Developer Groups
On Campus • Telkom University Bandung

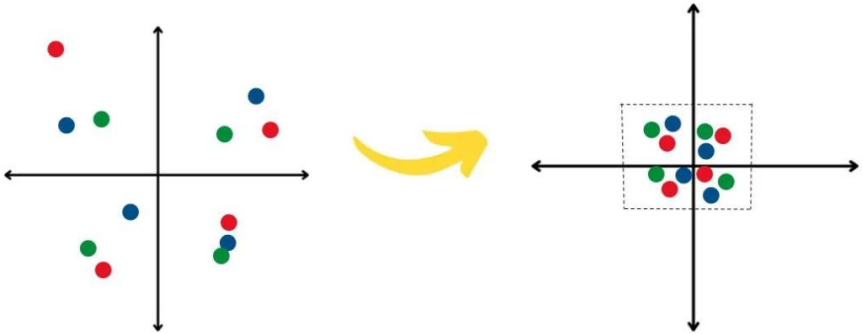
Data Preprocessing

Feature Scaling



Data Preprocessing:
Feature Scaling

Apa itu Feature Scaling?

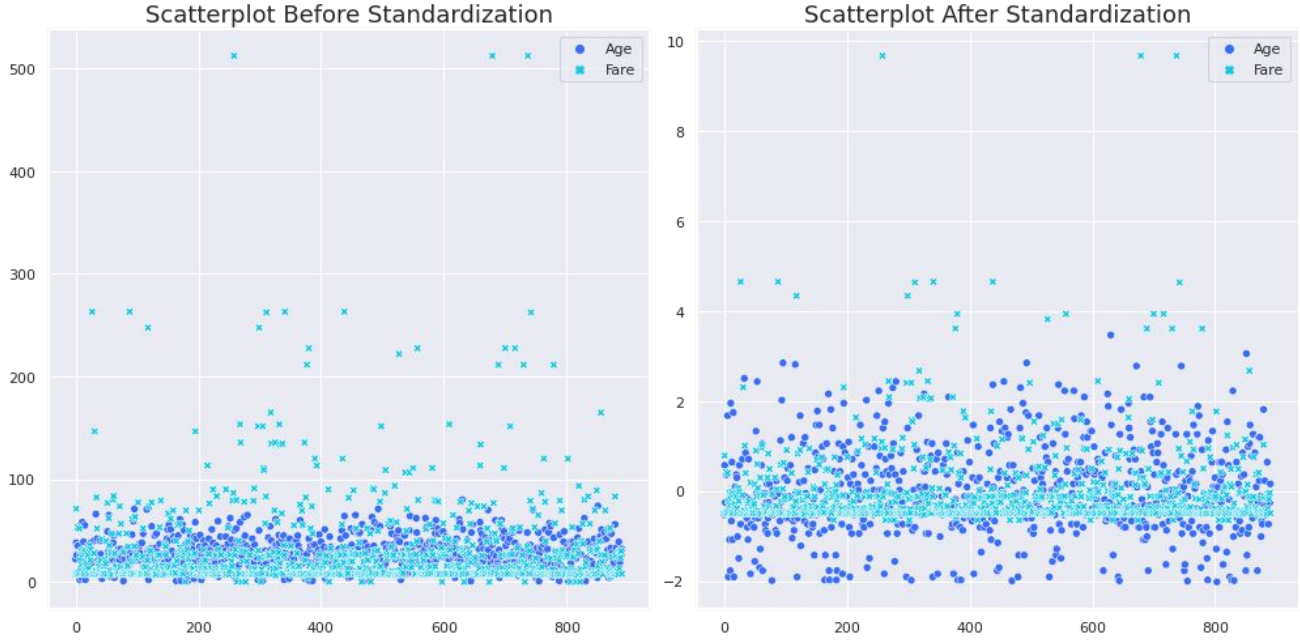


- **Feature Scaling** adalah langkah preprocessing di mana kita menstandarkan rentang variabel independen yang merupakan fitur dari dataset yang diberikan.
- **Why Feature Scaling?**
    - Hal ini penting karena fitur yang berbeda dapat memiliki skala yang sangat berbeda, dan banyak algoritma machine learning yang sensitif terhadap skala fitur (contoh: K-NN, SVM, Linear Regression).

https://python.plainenglish.io/understanding-feature-scaling-in-machine-learning-techniques-implementation-and-advantages-fd9065a349aa

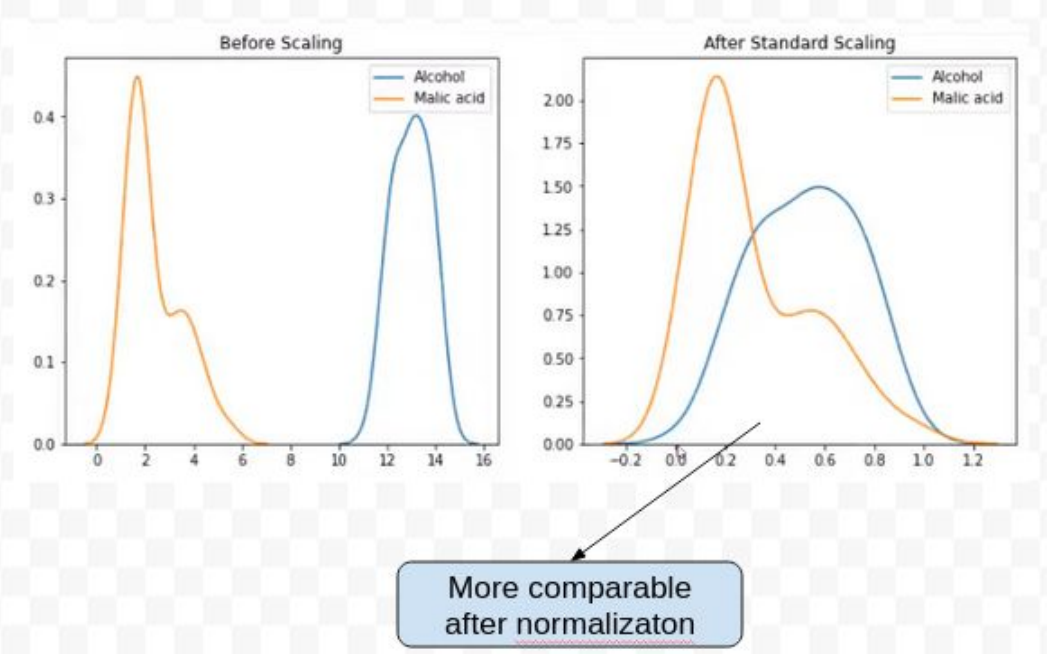
Data Preprocessing: Feature Scaling

Let's take an example of Feature Scaling



Data Preprocessing: Feature Scaling

Another Example



https://www.kaggle.com/discussions/getting-started/499532

Data Preprocessing:
Feature Scaling

Methods

- **Standardization**: mengubah fitur-fitur dataset Anda sehingga memiliki rata-rata 0 dan standar deviasi 1
- **Normalization**: mengubah fitur kolom numerik dalam kumpulan data untuk menggunakan skala yang sama, tanpa mendistorsi perbedaan dalam rentang nilai atau kehilangan informasi
    - Min-Max Scaling
    - Mean Normalization
    - Max Absolute Scaling
    - Robust Scaling

https://python.plainenglish.io/understanding-feature-scaling-in-machine-learning-techniques-implementation-and-advantages-fd9065a349aa

Data Preprocessing:
Feature Scaling

Standardization

$$X_{std} = \frac{X - \mu}{\sigma}$$

Dimana:

- X: original value,
- μ : mean dari fitur tersebut,
- σ : standar deviasi dari fitur tersebut.

Data Preprocessing:
Feature Scaling

Min-Max Scaling

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Dimana:

- X: original value,
- Xmin : nilai terkecil dari fitur tersebut,
- Xmax : nilai terbesar dari fitur tersebut.

Data Preprocessing:
Feature Scaling

What to use?

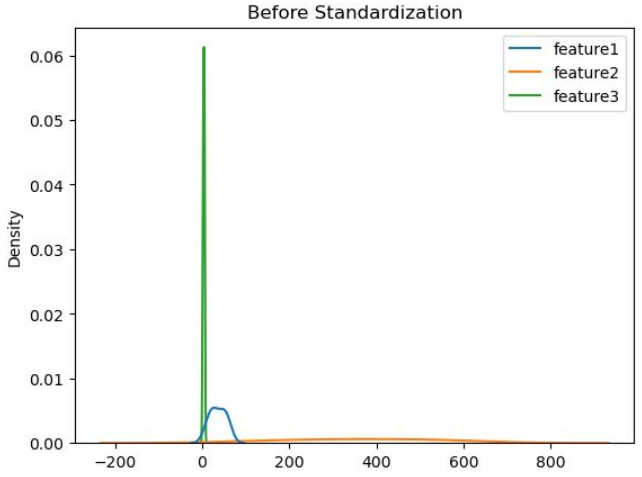
- **Standardization**: jika membutuhkan fitur yang dinormalisasi.
- **Min Max Scaling**: ketika fitur memiliki unit atau distribusi yang berbeda, dan Anda mengharapkan fitur tersebut mengikuti distribusi normal.
- **Mean Normalization**: digunakan dalam kasus-kasus di mana kita membutuhkan data terpusat.
    - Lebih bagus Standardization.
- **Max Absolute Scaling**: digunakan pada data yang jarang.
    - Data yang jarang berarti data yang mengandung jumlah nol maksimum
- **Robust Scaling**: ketika ada outlier dan dapat meng-handle rentang yang lebih besar.

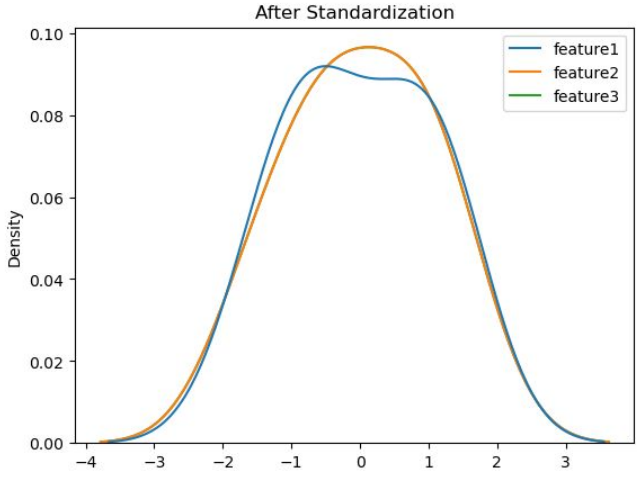
Rule of thumb: Gunakan Standardization untuk data berdistribusi normal, selain itu gunakan Min Max Scaling.

Data Preprocessing: Feature Scaling

Standardization

```python
# apply standardization
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

df_scaled = scaler.fit_transform(df)

df_scaled = pd.DataFrame(df_scaled, columns=df.columns) # add column names back
```

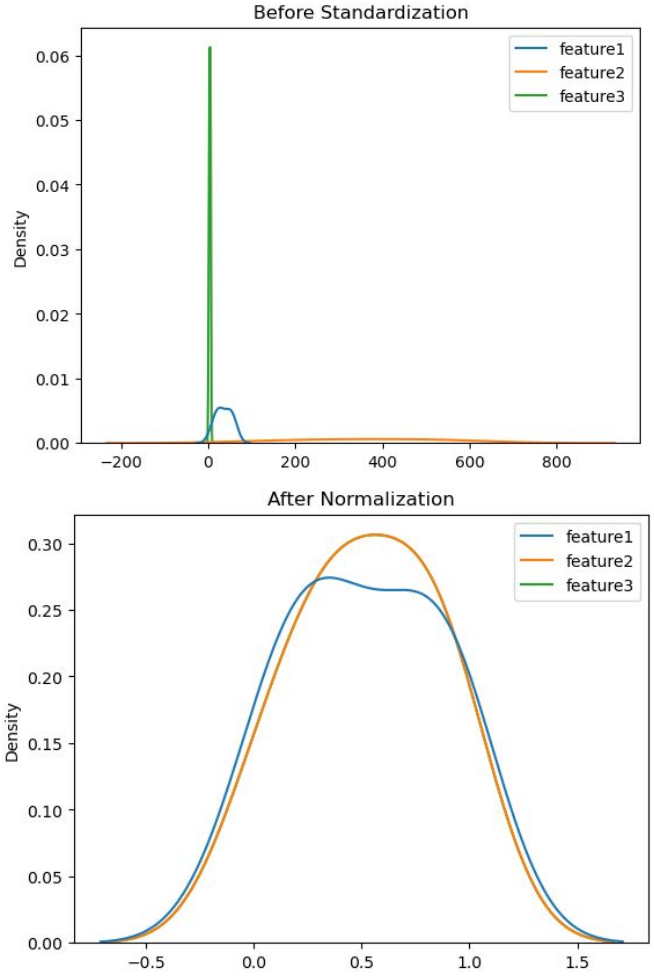




Data Preprocessing: Feature Scaling

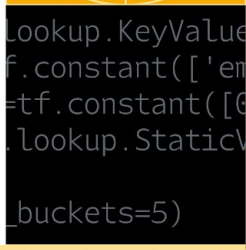
Min Max Scaling

```python
# apply standardization
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df_scaled = scaler.fit_transform(df)

df_scaled = pd.DataFrame(df_scaled, columns=df.columns) # add column names back
```



Google Developer Groups
On Campus • Telkom University Bandung

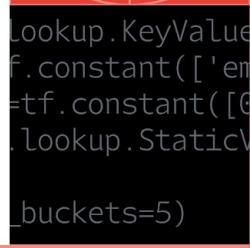
Hands On Part 1



Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering

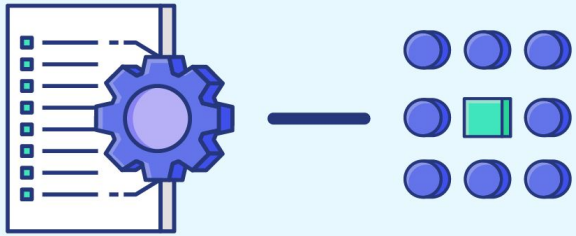
Definition



Feature Engineering: Definition

Apa itu Feature Engineering?

- **Feature Engineering** proses pembuatan fitur baru atau mengubah fitur yang sudah ada untuk meningkatkan kinerja model machine learning.
- **Tujuannya** untuk membuat data lebih sesuai dengan masalah yang dihadapi.



Feature Engineering: Definition

Kenapa butuh Feature Engineering?

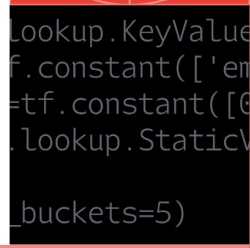
Jawabannya:

Tanpa feature engineering, model tidak bisa memahami pola data dengan baik, hasil prediksi jadi kurang akurat!

Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering

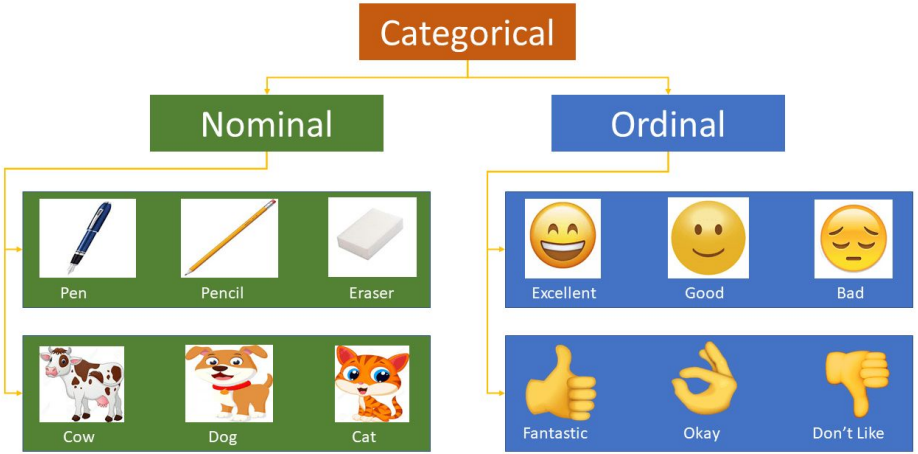
Categorical Encoding



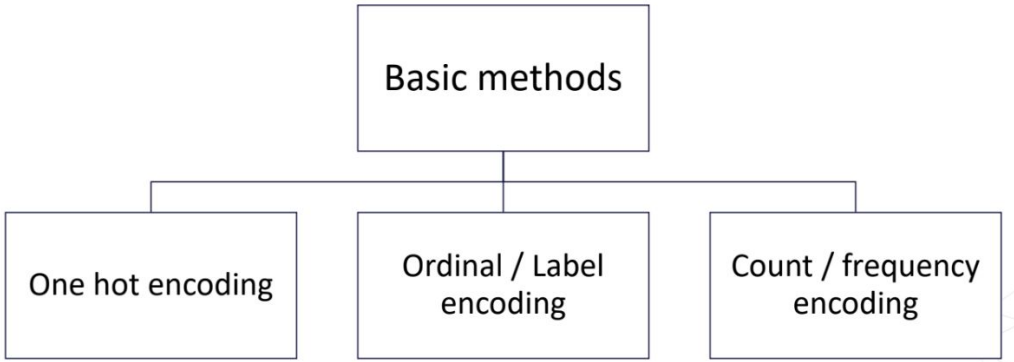
Feature Engineering: Categorical Encoding

Categorical Encoding

- **Categorical Encoding** mengacu pada penggantian string kategori dengan representasi numerik.
- **Untuk menghasilkan** variabel yang dapat digunakan untuk melatih model pembelajaran mesin.
- **Untuk membangun fitur prediktif dari kategori.**



Categorical Encoding



Feature Engineering: Categorical Encoding

One Hot Encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

Feature Engineering: Categorical Encoding

One Hot Encoding (k-1 variables)

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow
1	0
1	0
0	1
0	0
0	1

Feature Engineering: Categorical Encoding

Label Encoding

Height
Tall
Medium
Short

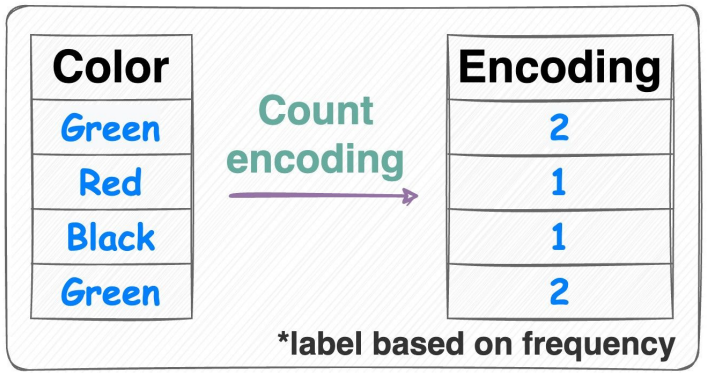
Height
0
1
2

Ordinal Encoding (order matters)

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

Feature Engineering: Categorical Encoding

Count Encoding



Frequency Encoding

Numerical value	Animal
1.5	cat
3.6	cat
42	dog
7.1	crocodile

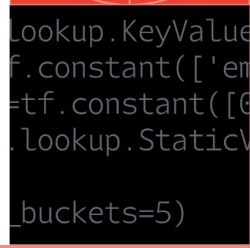
Frequency encoding

Numerical value	Animal_freq
1.5	0.5
3.6	0.5
42	0.25
7.1	0.25

Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering

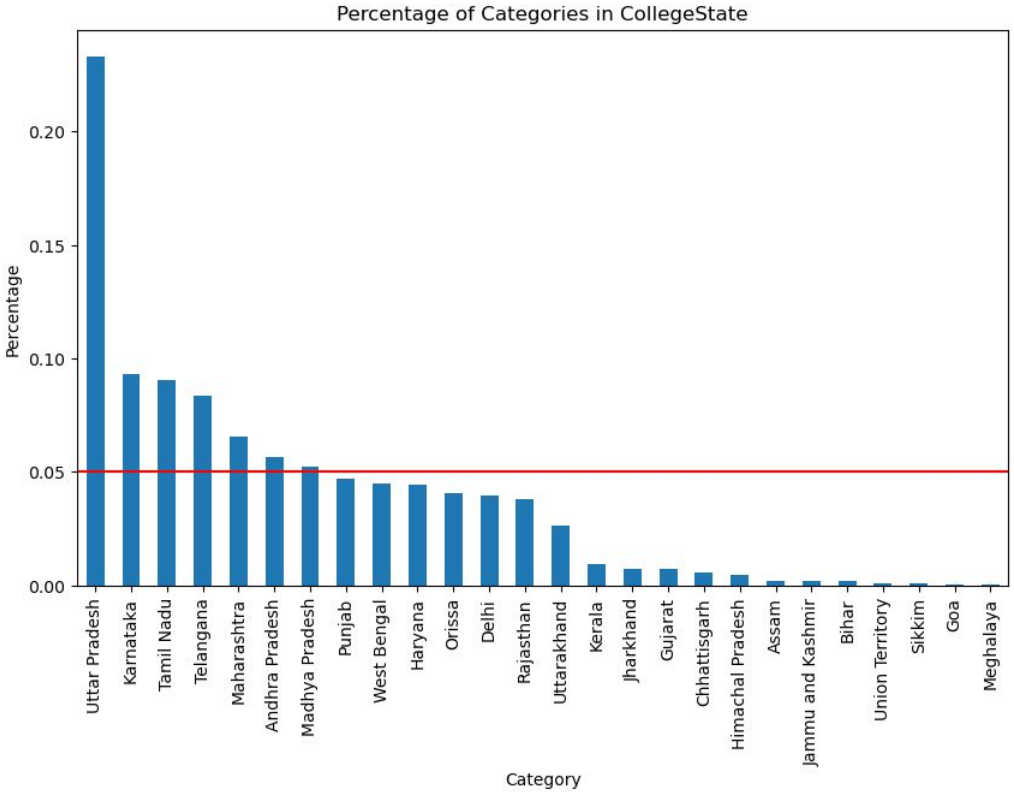
Rare Label



Feature Engineering: Rare Label

Rare Label

- **Rare Label** adalah adalah nilai / value kategorikal yang hanya muncul dalam proporsi kecil dari pengamatan dalam suatu dataset.
- **Problem**:
  - Possible **High Cardinality**.
  - Algoritma Machine Learning susah mencari pattern dari nilai yang rare.

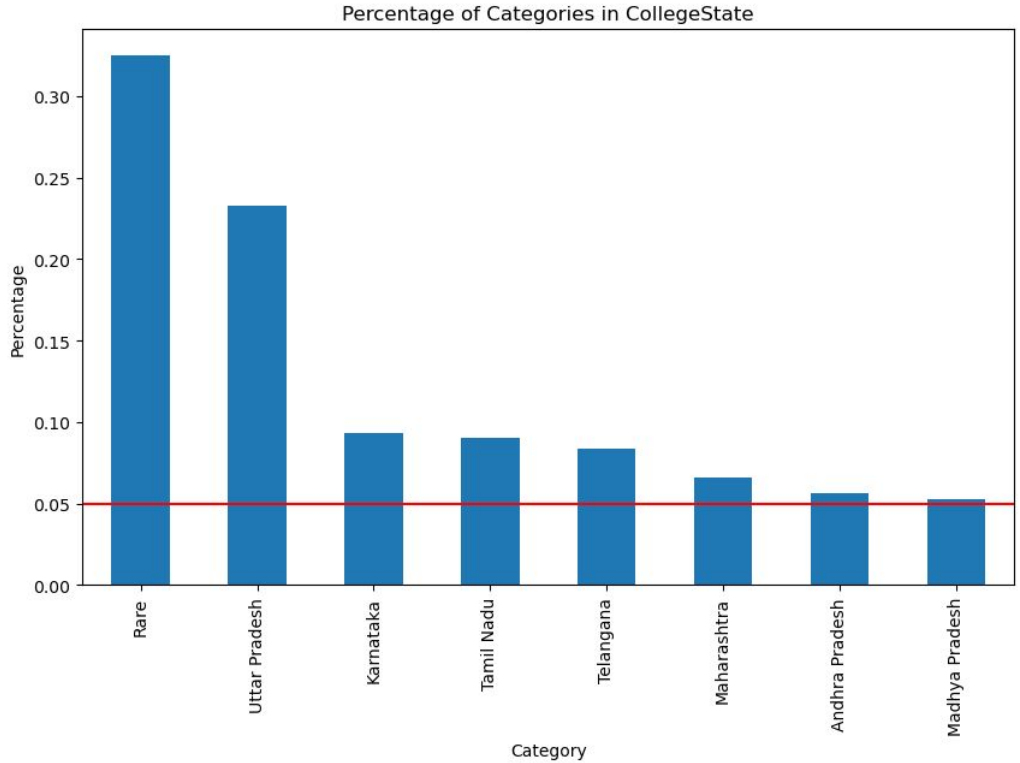


Ada 26 unique values = **High Cardinality**

Solution?

Gabungkan Rare Label menjadi 1 label atau value.

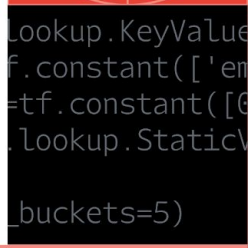
Setelah Penggabungan Rare Label



Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering

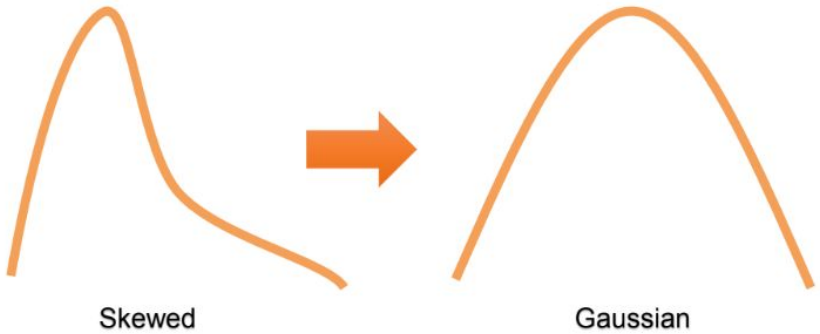
Variable Transformation



Feature Engineering: Variable Transformation

Variable Transformation

- Terkadang, kita dapat memperbaiki kegagalan dalam asumsi dengan mengubah variabel sebelum melakukan analisis. (contoh: outliers)
- Hal ini akan **meningkatkan performa** model machine learning.

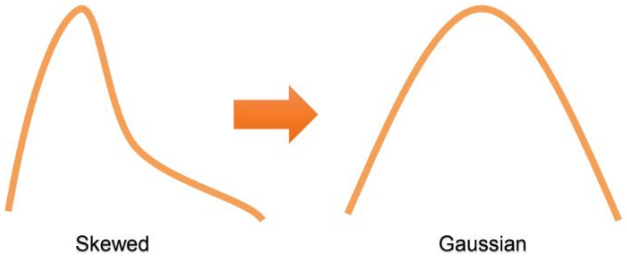


Feature Engineering: Variable Transformation

Which variables can we transform?

- Kita dapat mentransformasi variabel target itu sendiri ketika distribusinya miring (skewed).
- **Mentransformasi variabel prediktor (yang ingin di prediksi), sering kali membantu memenuhi asumsi model ketika data mentah tidak.**

← Machine Learning



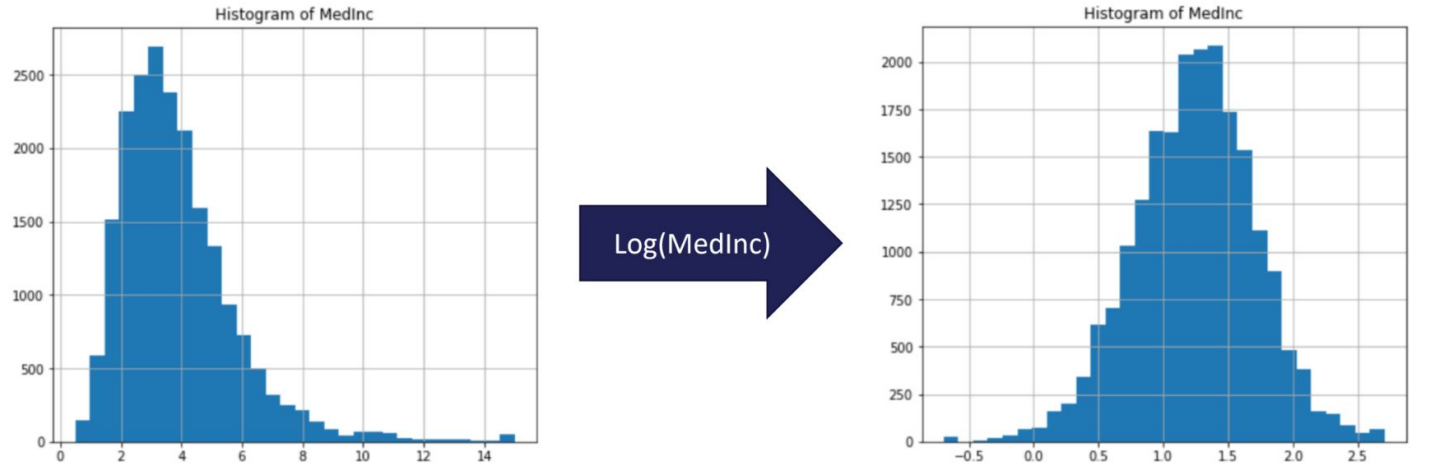
Feature Engineering: Variable Transformation

Methods:

- **Log Transformation**
  - Cocok kalau: data **bernilai positif** dengan **distribusi miring ke kanan** (right-skewed distribution)
  - Rumus: X_new = log(X)
- **Reciprocal**
  - Cocok kalau: data memiliki rasio, yaitu nilai yang dihasilkan dari pembagian dua variabel.
  - Contoh umum: Kepadatan penduduk, yaitu jumlah orang per area, atau hunian rumah, yaitu jumlah penghuni per rumah.
  - Rumus: X_new = 1 / X
- **Square-root**
  - Cocok kalau: variabel dengan distribusi Poisson (counts).
  - Rumus: X_new = √X
- **Many Other!**

Feature Engineering: Variable Transformation

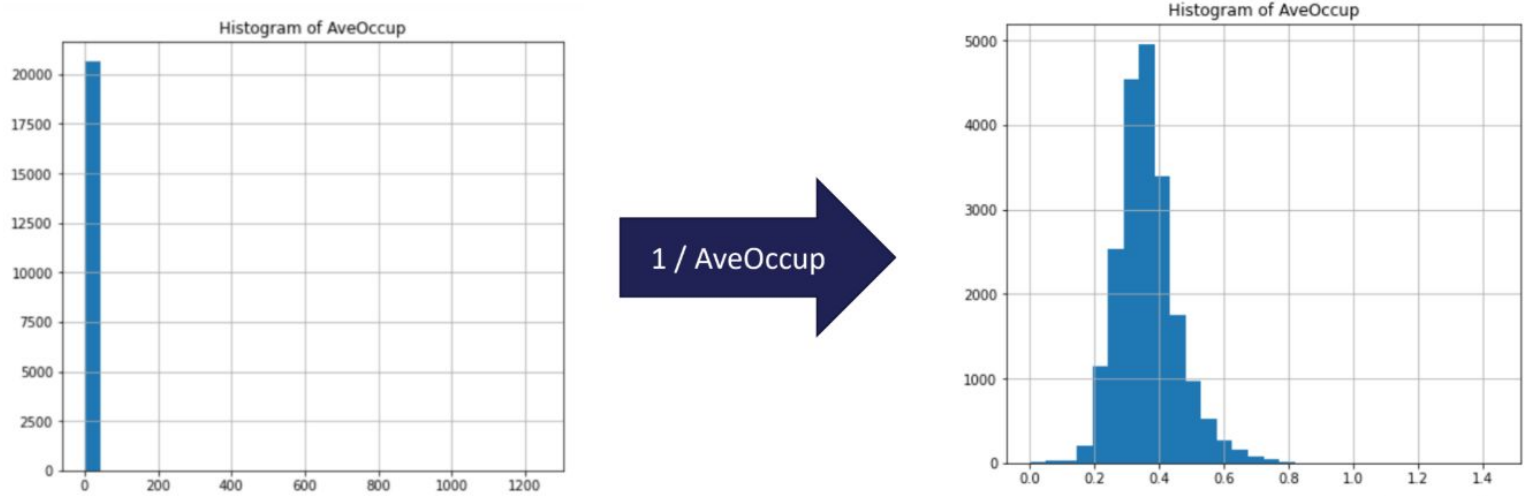
Log Transformation



California housing dataset.

Feature Engineering: Variable Transformation

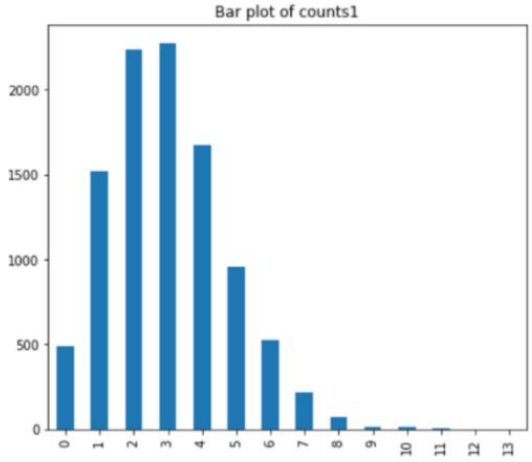
Reciprocal

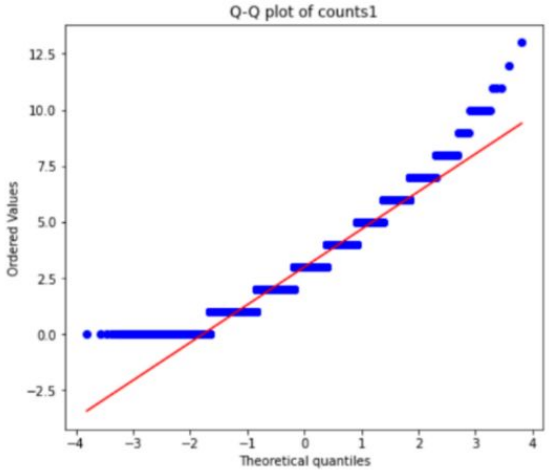


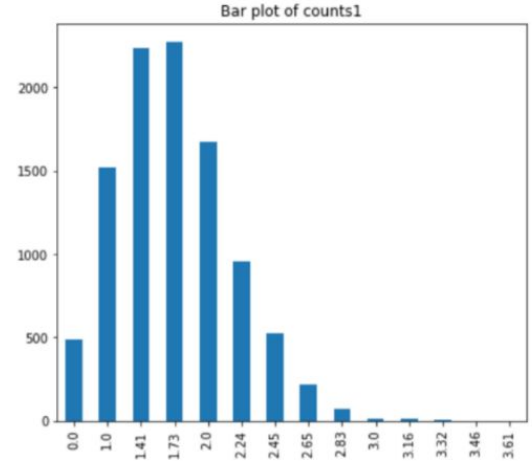
California housing dataset.

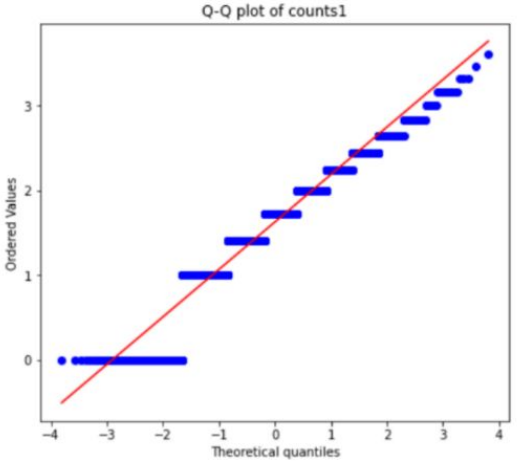
Feature Engineering: Variable Transformation

Square-root





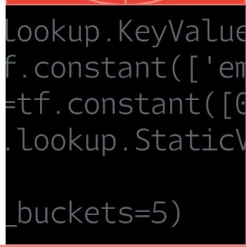




Google Developer Groups
On Campus • Telkom University Bandung

Feature Engineering

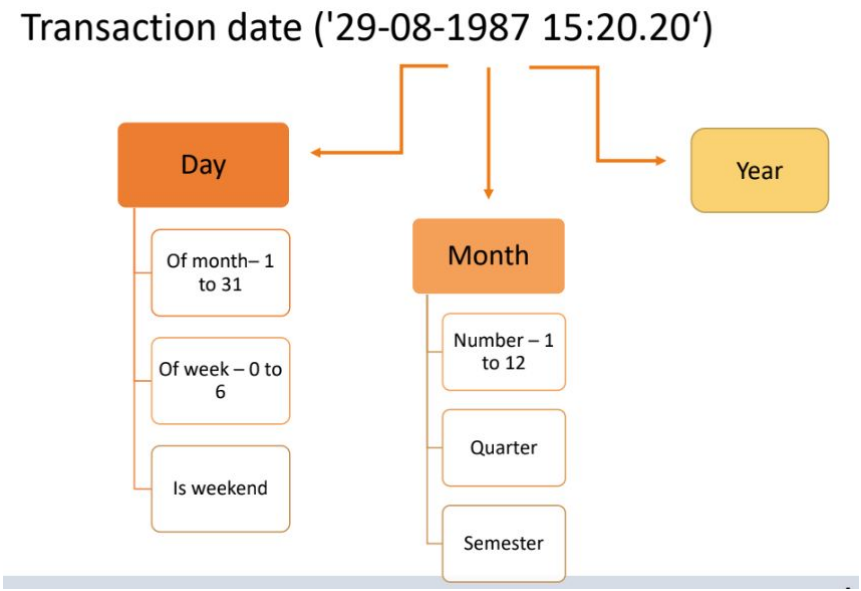
Date Time Feature



Feature Engineering: Date Time Feature

Date Time Feature

- Tanggal dan waktu, atau variabel **datetime**, mereka mengambil tanggal dan/atau waktu sebagai nilai.
  - Date of birth ('29-08-1987', '12-01-2012')
  - Date of application ('2016-Dec', '2013-March')
  - Time of accident (12:20:45)
  - Payment date ('29-08-1987 15:20.20‘)



Feature Engineering: Date Time Feature

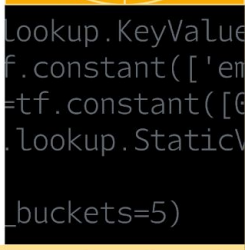
Date Time Feature

Transaction date ('29-08-1987 15:20.20')

Date	First / last of year
	First / last of quarter
	Leap year
	Week of year

Google Developer Groups
On Campus • Telkom University Bandung

Hands On Part 2



Any Question ?

Google Developer Groups

Let’s Connect!

- **Instagram: @rdavaa_**
- **LinkedIn:** www.linkedin.com/in/ardava-barus