

Data Cleansing for Machine Learning

Rakamin Trial Class



Saya



Pararawendy Indarjo

Email : pararawendy19@gmail.com

Linkedin : <https://www.linkedin.com/in/pararawendy-indarjo/>

Blog : medium.com/@pararawendy19



**Universiteit
Leiden**

Master of Science

Leiden University (2017-2019)

Focused on application of theoretical machine learning & reinforcement learning



Senior Data Scientist

Bukalapak (2020 - present)





Hafizh Adi P

Data Scientist



Halo!



東京工業大学

Tokyo Institute of Technology

Master of Eng. - Artificial Intelligence
(2017 - 2019)



Data Scientist (2019 - now)

Greetings!



Fiqry Revadiansyah

Sr. Data Scientist



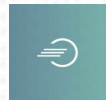
Sr. Data Scientist

Ajaib.co.id



Data Science - AAI

Accenture Indonesia



RnD Manager Data Science

Purwadhika Digital Technology School



Data Science

Bukalapak.com



LINKEDIN



SPEAKERDECK

Sesi ini:

1. Mengapa data perlu dipreproses?
2. *Missing data*
3. *Duplicated data*
4. *Outliers*
5. *Feature encoding*
6. QnA



Hands-On Required :

Hands - On :

Hands-On - Data Cleansing.ipynb

Dataset :

1. **botak.csv**

**Klik disini untuk mengakses
folder Database**

Isi daftar hadir di sini!
bit.ly/RTCDS24DaftarHadir3



Mengapa Data perlu Dipreproses?

Kebersihan data adalah sebagian dari sukses



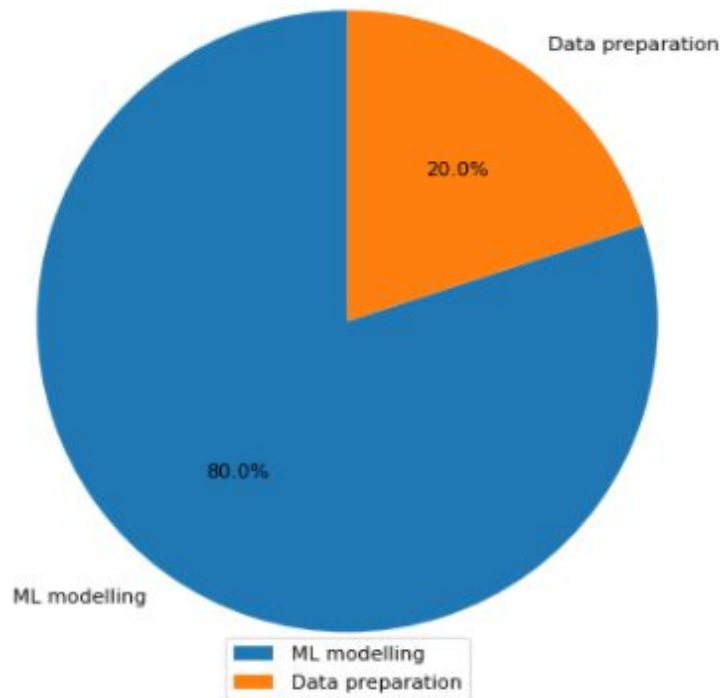
Seberapa kotorkah data-data di dunia nyata? Sangat kotor!

Contoh penyebab ketidakbersihan data:

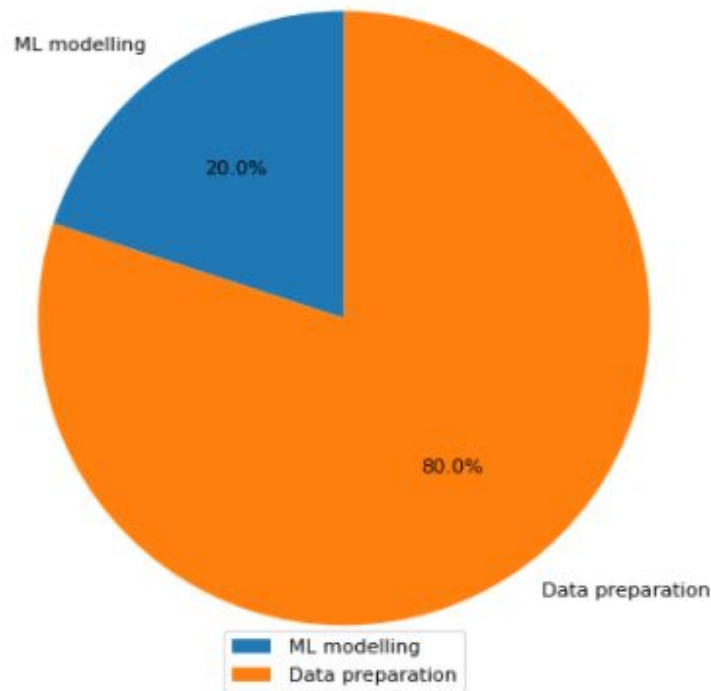
- User asal/salah mengisi data
- Jenis input yang tidak wajib
- Kesalahan implementasi *tracker data/engineering* mistakes
 - *Scammer, abuser*
 - Etc...



Ekspektasi :)



Realita :(



Data Pre-processing



Dataset

botak.csv

- **Deskripsi:**

Dataset sintetik. Memprediksi peluang botaknya seseorang dari beberapa atribut mengenai orang tersebut.

- **Data:**

Setiap baris mewakili satu orang, setiap kolom berisi atribut orang tersebut.

**ilustrasi tidak ada hubungannya dengan data*



Photo of bald dirty mad man on gray background



Cuplikan Datanya...

- Misalkan kita namakan data botak kita sebagai “df”

```
df.head()
```

	umur	jenis_kelamin	pekerjaan	provinsi	gaji	is_menikah	is_keturunan	berat	tinggi	sampo
0	42.0	Laki-laki	PNS	Padang	7.864005e+06	0	0.0	61.928685	152.174164	Deadbuoy
1	33.0	Laki-laki	Pegawai swasta	Palangkaraya	6.492662e+06	0	0.0	49.374507	152.822969	Moonsilk
2	59.0	Laki-laki	Freelance	Serang	8.086303e+06	1	0.0	59.072807	159.911047	Merpati
3	38.0	Laki-laki	Pegawai swasta	Gorontalo	1.147321e+07	0	1.0	68.338014	162.558997	Moonsilk
4	33.0	Perempuan	Freelance	Tanjungseler	1.475942e+07	0	0.0	53.769996	154.579510	Pantone



Missing Data

1: Data yang Hilang



Berapa banyak data yang hilang?

```
df.isna().sum()
```

```
umur          0
jenis_kelamin  9
pekerjaan     67
gaji          23
is_menikah    0
is_keturunan  15
berat         39
tinggi        0
sampo        57
is_merokok    0
pendidikan    0
botak_prob    0
dtype: int64
```

Cara mengecek jumlah nilai yang hilang pada dataframe:

- `df.isna().sum()`



Teknik #1: Hapus (drop) baris-baris dengan data yang hilang

Ketika kita punya cukup banyak data dan jumlah data yang hilang tidak signifikan, biasanya cukup hapus baris-baris dengan data yang hilang.

```
df = df.dropna()
```

```
df.dropna(inplace=True)
```

Penghapusan dapat dilakukan dengan fungsi `df.dropna()`. Kode di atas menunjukkan contoh 2 cara berbeda menggunakan `df.dropna()`.

Berikut penjelasan untuk 2 parameter yang dipakai di atas:

- `inplace`: Nilai `True` atau `False`.
- Apabila `True`, tidak mengembalikan dataframe baru tapi langsung menghapus di dataframe awal.



Teknik #2: Isi data-data yang kosong/Imputation

Ketika kita tidak mau menghapus satu pun baris data, kita bisa mengisi kekosongan data secara manual.

Kita isi dengan apa? Biasanya ada 2 hal yang dipertimbangkan:

- Konteks masalah: nilai apa yang paling masuk akal?
- Performa model ML: nilai apa yang menghasilkan performa model tertinggi?

Pengisian dapat dilakukan dengan fungsi `df.fillna()`

Find out in the
Bootcamp :)



Duplicated Data

2: Data yang Sama/Berulang



Berapa banyak data yang duplikat?

```
1 df.duplicated().sum()
```

```
0
```

Pengecekan jumlah nilai yang duplikat pada dataframe dapat dilakukan dengan `df.duplicated().sum()`.

Ketika kita yakin kita tidak memerlukan baris-baris duplikat, biasanya cukup hapus baris-baris tersebut.

```
df = df.drop_duplicates()
```

```
df.drop_duplicates(inplace=True)
```



Outliers

3: Data yang Berbeda (jauh)



Outlier itu apa?

Outlier adalah data point (baris) yang nilainya ekstrem/jauh berbeda dari data-data lain pada umumnya. Bisa muncul dari:

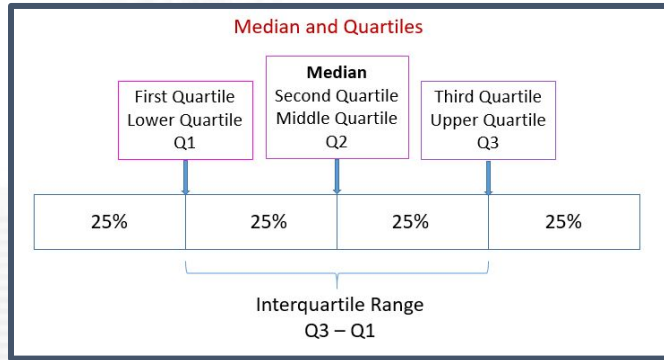
- Kesalahan pada pengambilan data
- Keberadaan individu-individu yang 'spesial'

Kenapa outlier jadi masalah?

Karena dapat menyebabkan model machine learning kita berperforma buruk



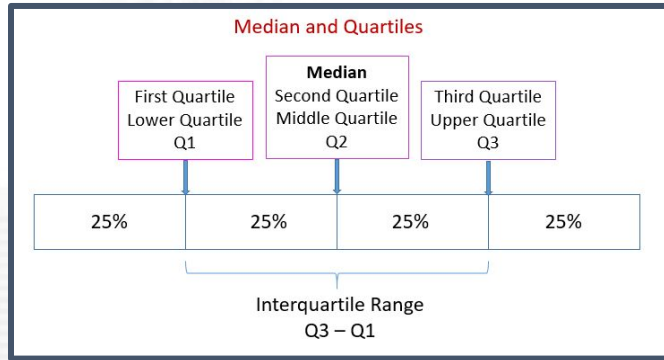
Refresher: *IQR (inter quartile range)*



IQR: lebar Q3-Q1



Definisi outlier berdasarkan *IQR (inter quartile range)*



IQR: lebar $Q3 - Q1$

Outlier: Data yang jauh lebih ekstrim dari $Q1$ atau $Q3$

- Seberapa jauh?
 - 1.5 kali IQR



Menghapus outlier berdasarkan IQR

```
1 Q1 = df['umur'].quantile(0.25)
2 Q3 = df['umur'].quantile(0.75)
3 IQR = Q3 - Q1
4 low_limit = Q1 - (1.5 * IQR)
5 high_limit = Q3 + (1.5 * IQR)
6 filtered_entries = ((df['umur'] >= low_limit) & (df['umur'] <= high_limit))
7 df = df[filtered_entries]
```

Kode di atas menunjukkan cara menghapus baris berdasarkan outlier di kolom umur menggunakan IQR. Berikut penjelasannya:

1. Hitung Q1
2. Hitung Q3
3. Hitung IQR
4. Hitung batas bawah untuk outlier
5. Hitung batas atas untuk outlier
6. Buat filter boolean berdasarkan apakah nilai di bawah batas bawah atau di atas batas atas
7. Ambil baris data saat datanya ada di antara batas bawah dan atas



Feature Encoding

Mengakomodasi feature categorical



***Feature Encoding* itu apa?**

***Feature Encoding* adalah proses mengubah feature categorical menjadi feature numeric.**



Data Besaran Penghasilan dari Survei Abhal²

Gender	Pendidikan	Pekerjaan	Penghasilan (juta)
Laki-laki	S1	SWASTA	7
Laki-laki	SMA	PNS	13
Perempuan	S1	PNS	15
Laki-laki	S2	FREELANCE	24
Perempuan	S3	PNS	17
Perempuan	S1	SWASTA	23
Perempuan	SMA	FREELANCE	12

Pertanyaan

Bagaimana cara mengubah fitur dalam bentuk **STRING** menjadi angka?

- Sebab Python hanya 'mengerti' fitur numerik untuk membangun model



Teknik #1: Label Encoding

Label Encoding adalah perubahan feature categorical menjadi numeric dengan memberikan angka yang berbeda bagi masing-masing nilai unik

```
mapping_gender = {
    'Laki-laki': 0,
    'Perempuan': 1
}
df['gender'] = df['gender'].map(mapping_gender)

mapping_pendidikan = {
    'SMA': 0,
    'S1': 1,
    'S2': 2,
    'S3': 3
}
df['pendidikan'] = df['pendidikan'].map(mapping_pendidikan)
```



	gender	pendidikan	pekerjaan	penghasilan
0	0	1	SWASTA	7
1	0	0	PNS	13
2	1	1	PNS	15
3	0	2	FREELANCE	24
4	1	3	PNS	17
5	1	1	SWASTA	23
6	1	0	FREELANCE	12



Lalu bagaimana dengan kolom 'pekerjaan'?

Teknik #2: One-hot Encoding

One-hot encoding adalah perubahan feature categorical menjadi numeric dengan menjadikan masing-masing nilai unik feature tersendiri

```
1 pd.get_dummies(df['pekerjaan'], prefix='kerja')
```

Kode di atas menunjukkan cara melakukan one-hot encoding pada kolom pekerjaan menggunakan `get_dummies()`. Berikut penjelasannya:

1. Parameter pertama adalah kolom yang ingin di one-hot encoding (pekerjaan)
2. Parameter `prefix` diisi dengan nama awalan dari kolom-kolom baru yang akan dihasilkan
3. Fungsi ini akan mengembalikan dataframe baru yang berisi feature-feature numerik



Teknik #2: One-hot Encoding (lanjutan)

```
1 pd.get_dummies(df['pekerjaan'], prefix='kerja')
```

	kerja_FREELANCE	kerja_PNS	kerja_SWASTA
0	0	0	1
1	0	1	0
2	0	1	0
3	1	0	0
4	0	1	0
5	0	0	1
6	1	0	0

Ketika kita tampilkan dataframe yang dihasilkan terlihat bahwa setiap nilai unik berubah menjadi kolom baru. Awalan nama kolom-kolom baru ini sesuai dengan isi parameter `prefix`.



Data Besaran Penghasilan dari Survei Abhal² (encoded)

gender	pendidikan	kerja_freelance	kerja_PNS	kerja_swasta	penghasilan
0	1	0	0	1	7
0	0	0	1	0	13
1	1	0	1	0	15
0	2	1	0	0	24
1	3	0	1	0	17
1	1	0	0	1	23
1	0	1	0	0	12

Hore! Semua fitur sudah numerik!



Hands-On Session :

Hands - On :

Hands-On - Data Cleansing.ipynb

Dataset :

1. botak.csv



Sudah.

Sesi tanya-jawab

Isi daftar hadir di sini!
bit.ly/RTCDS24DaftarHadir3





Register Now, and Get Closer to Your Dream Job

Chat Data Science Admission Now!