

Virtual Internship Experience

Data Cleaning

Disclaimer

“Dokumen ini memiliki hak cipta. Barang siapa yang menyebarluaskan atau menduplikasi tanpa izin dari instansi terkait dapat diproses sesuai dengan ketentuan hukum yang berlaku.”

Outline

- 1** **Data *Profiling***
- 2** **Data *Imputation***
- 3** **Data *Transformation***
- 4** **Studi Kasus**

1 Data *Profiling*

Data Profiling mengacu pada proses memeriksa dan menganalisis kumpulan data untuk mendapatkan informasi tentang struktur, konten, dan kualitas data.

Ini melibatkan eksplorasi karakteristik dataset (EDA), seperti ukurannya, jenis data, distribusi, pola, dan nilai-nilai yang hilang, untuk memahami lanskap data secara keseluruhan.



1 Data Profiling

Implementasi

❏ .describe()

digunakan untuk menghasilkan statistik deskriptif untuk DataFrame atau Series. Fungsi ini menghitung hal-hal berikut:

- Count: Jumlah nilai yang tidak kosong,
- Rata-rata: Nilai rata-rata (mean).
- Std: Standar deviasi.
- Min: Nilai minimum.
- 25%: Persentil ke-25 (juga dikenal sebagai kuartil pertama).
- 50%: Persentil ke-50 (juga dikenal sebagai median).
- 75%: Persentil ke-75 (juga dikenal sebagai kuartil ketiga).
- Max: Nilai maksimum.

Index	rand_num	count	5
0	7	mean	4.4
1	1	std	2.701
2	6	min	1
3	2	25%	2
4	6	50%	6
		75%	6
		max	7

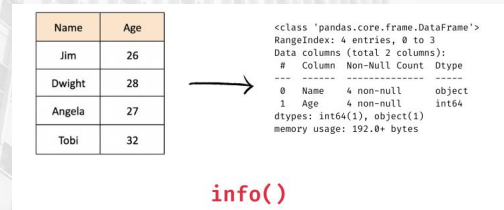
Gambar 1. contoh hasil describe pada data (kanan)

1 Data Profiling

Implementasi

□ .info()

Metode `pandas.info()` digunakan untuk mencetak ringkasan singkat dari `DataFrame`. Metode ini mencetak informasi tentang `DataFrame` termasuk tipe indeks dan kolom, data yang tidak kosong, dan penggunaan memori.



Name	Age
Jim	26
Dwight	28
Angela	27
Tobi	32

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0  Name    4 non-null      object
1  Age     4 non-null      int64
dtypes: int64(1), object(1)
memory usage: 192.0+ bytes
```

info()

Gambar 2. contoh hasil fungsi `.info()`

□ .head()

Untuk mendapatkan tampilan data, dapat menggunakan function `.head()` untuk mengembalikan `n` baris pertama dari `DataFrame`. Secara default, `n` diatur menjadi 5. Metode `head()` berguna untuk dengan cepat melihat data dalam `DataFrame`.

```
df.head(8) =>
```

	Anime	Episodes	Year
0	One Piece	1009	1999
1	Naruto	720	2002
2	Bleach	366	2004
3	Hunter X Hunter	148	2011
4	Attack On Titan	74	2013
5	Gintama	366	2006
6	Code Geass	50	2007
7	Death Note	37	2008

Gambar 3. contoh hasil fungsi `.head()`

2 Data Imputation

Data Imputation adalah metode untuk menggantikan data yang hilang (missing value) dengan nilai pengganti. Hal ini digunakan untuk mempertahankan sebagian besar data dan informasi dalam kumpulan data. Metode ini digunakan karena tidak praktis untuk menghapus data dari setiap kumpulan data setiap kali terjadi. Selain itu, melakukannya secara substansial akan mengurangi ukuran kumpulan data, menimbulkan pertanyaan tentang bias dan mengganggu analisis.

Terdapat banyak metode pengisian data yang berbeda, masing-masing dengan keuntungan dan kerugiannya sendiri. Beberapa metode paling umum meliputi:

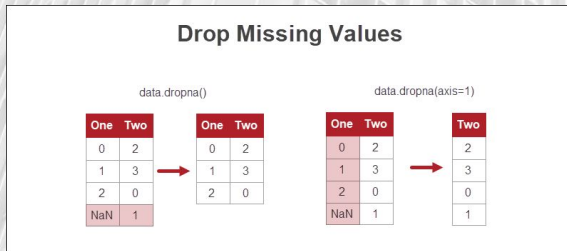
- Menghapus data
- Random Sample Imputation
- Mean Imputation
- Median Imputation
- Mode Imputation



2 Data Imputation

❑ Menghapus data

- + Menghapus nilai yang hilang dapat meningkatkan akurasi analisis Anda.
- + Menghapus nilai yang hilang juga dapat membuat data Anda lebih mudah untuk dianalisis.
- Menghilangkan data dengan nilai yang hilang dapat mengurangi ukuran dataset Anda.
- Menghilangkan data dengan nilai yang hilang juga dapat memperkenalkan bias ke dalam data Anda.



Gambar 1. contoh implementasi penghapusan data pada baris (kiri) dan pada kolom (kanan)

❑ Random Sample Imputation

- + sederhana dan mudah diimplementasikan.
- + Tidak memerlukan perangkat library khusus atau keahlian tertentu.
- + Dapat digunakan untuk mengisi nilai-nilai yang hilang untuk jenis variabel apa pun.
- + Relatif tidak bias.
- Dapat memperkenalkan noise ke dalam data.
- Dapat meremehkan variansi data.
- Dapat kurang akurat dibandingkan dengan metode imputasi lainnya

2 Data Imputation

☐ Mean Imputation

- + Mudah untuk dihitung
- + Terhitung cepat dari segi waktu implementasi
- + Dapat digunakan dengan berbagai jenis variabel
- Peka terhadap pencilan data (*outliers*)
- Dapat merusak distribusi data
- Dapat mengabaikan variansi data

```
df.fillna(df.mean())
```

Gambar 2. mengisi nilai kosong dengan mean pada pandas

☐ Median imputation

- + Lebih kurang sensitif terhadap *outliers* daripada imputasi rata-rata
- + Tidak merusak distribusi data sebanyak *Mean Imputation*
- + Dapat digunakan dengan jenis variabel apa pun
- Lebih mahal secara komputasional dibandingkan dengan *mean imputation*

```
df.fillna(df.median())
```

Gambar 3. mengisi nilai kosong dengan median pada pandas

☐ Mode Imputation

- + Sangat mudah dalam kalkulasi
- + Waktu implementasi tergolong cepat
- Mudah terpengaruh oleh outlier
- Dapat merusak distribusi data
- Dapat mengestimasi terlalu tinggi variansi data

```
df['salary'] = df['salary'].fillna(df['salary'].mode()[0])
```

Gambar 4. mengisi nilai kosong dengan mode pada pandas

salary	salary
270000.0	270000.0
200000.0	200000.0
250000.0	250000.0
NaN	300000.0
425000.0	425000.0

Gambar 5. contoh sebelum dan sesudah mengisi missing values dengan modus (mode)

3

Data Transformation

Data Transformation adalah proses mengkonversi data dari satu format ke format lainnya. Berikut adalah beberapa manfaat dari transformasi data:

- Meningkatkan kualitas data
- Meningkatkan kegunaan data
- Meningkatkan keamanan data
- Mengurangi biaya penyimpanan data



3 Data Transformation

Implementasi

□ .astype()

Metode `astype()` dalam Pandas digunakan untuk mengubah tipe data dari sebuah kolom. Sintaks untuk metode `astype()` adalah sebagai berikut:

The name of the dataframe column

`your_dataframe.astype({'column': 'datatype'})`

The datatype you want to use for that column
(e.g., category, int8, float16, etc)

Gambar 1. syntax untuk fungsi `astype()`

```
[3]: import pandas as pd
dat = {'Gender': ['M', 'M', 'M', 'F', 'M', 'F', 'M'], 'NAME': ['Karlos', 'Gaurav', 'Ray', 'Dee', 'Steve', 'Su', 'Ganesh']}
b = pd.DataFrame(dat)
print(" Give Data and their type is: \n")
print(b)
b.dtypes
b['Gender'] = b['Gender'].astype('category')
b.dtypes

Give Data and their type is:
Gender  NAME
0      M  Karlos
1      M  Gaurav
2      M    Ray
3      F    Dee
4      M  Steve
5      F    Su
6      M  Ganesh

[3]: Gender  category
NAME      object
dtype: object
```

Gambar 2. contoh implementasi fungsi `astype()`

4 Studi Kasus

Sebagai seorang Data Scientist Kalbe Nutritionals, anda diminta untuk membuat model yang dapat memprediksi penjualan produk kalbe pada bulan depan. anda diberikan data invoice dan anda ingin melakukan eksplorasi data terlebih dahulu agar nantinya dapat membuat model dengan data yang berkualitas

Petunjuk Pengerjaan:

1. Anda dapat mengerjakan menggunakan *jupyter notebook* atau [google colab](#)
2. Lakukan data profiling dengan melakukan eksplorasi data terlebih dahulu dengan mengetahui deskriptif statistik data, tampilan data, dan informasi singkat mengenai data
3. Apabila ada *missing values*, terapkan imputasi yang sesuai
4. Pastikan semua data sudah menggunakan tipe data yang tepat.

[Link Dataset](#)

Step 1. Import libraries dan memuat data

```
import pandas as pd

df = pd.read_csv('Dataset CCV 4 - Data Cleaning.csv')
```

Step 2. Melakukan data profiling

Step 2.1 Melihat tampilan data

```
df.head()
```

	id_order	id_produk	nama_produk	kategori_produk	tanggal_pembelian	kuantitas	total_harga	PPN	bayar_cash	metode_bayar
0	1	3	hic1000 vitamin lemon	beverages	2023-04-09	5.0	39452.0	0.1	1.0	cash
1	2	6	nutrive benecol	special needs	2023-05-01	1.0	33732.0	0.1	1.0	cash
2	3	6	nutrive benecol	special needs	2023-03-22	NaN	23035.0	0.1	1.0	cash
3	4	5	prenagen lova	woman	2023-04-12	2.0	31237.0	0.1	0.0	link aja
4	5	6	nutrive benecol	special needs	2023-01-20	5.0	32000.0	0.1	1.0	cash

Step 2.2 Melihat informasi data

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1712 entries, 0 to 1711
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   id_order              1712 non-null  int64  
 1   id_produk             1712 non-null  int64  
 2   nama_produk           1712 non-null  object  
 3   kategori_produk       1712 non-null  object  
 4   tanggal_pembelian     1712 non-null  object  
 5   kuantitas              1692 non-null  float64 
 6   total_harga           1692 non-null  float64 
 7   PPN                   1711 non-null  float64 
 8   bayar_cash            1704 non-null  float64 
 9   metode_bayar          1712 non-null  object  
dtypes: float64(4), int64(2), object(4)
memory usage: 133.9+ KB
```

Step 2.3 Mendapatkan informasi karakteristik data

```
df.describe()
```

	id_order	id_produk	kuantitas	total_harga	PPN	bayar_cash
count	1712.000000	1712.000000	1692.000000	1692.000000	1.711000e+03	1704.000000
mean	856.500000	3.442173	3.947991	43542.563830	1.000000e-01	0.601526
std	494.356147	1.740345	1.983287	14971.400255	1.388185e-17	0.489728
min	1.000000	1.000000	1.000000	17264.000000	1.000000e-01	0.000000
25%	428.750000	2.000000	2.000000	30973.750000	1.000000e-01	0.000000
50%	856.500000	3.000000	4.000000	43339.500000	1.000000e-01	1.000000
75%	1284.250000	5.000000	6.000000	56225.750000	1.000000e-01	1.000000
max	1712.000000	6.000000	7.000000	69824.000000	1.000000e-01	1.000000

***Pada solusi, digunakan google colab.**

Step 3. Mengatasi missing values

Step 3.1 Imputasi

```
df = df.fillna(  
    {'kuantitas':float(f'{df.kuantitas.median()}' ),  
     'total_harga':float(f'{df.total_harga.median()}' ),  
     'PPN':0.1}  
)
```

Step 3.2 Drop Data

```
df = df.drop(columns = 'bayar_cash')  
  
df.isna().sum()  
  
id_order      0  
id_produk     0  
nama_produk   0  
kategori_produk 0  
tanggal_pembelian 0  
kuantitas      0  
total_harga    0  
PPN            0  
metode_bayar   0  
dtype: int64
```

Step 4. Melakukan transformasi data

```
df = df.astype({'tanggal_pembelian':'datetime64[ns]',  
               'kategori_produk':'category',  
               'kuantitas':'int64',  
               'total_harga':'int64',  
               })  
  
df.info()  
  
<class 'pandas.core.frame.DataFrame'  
RangeIndex: 1712 entries, 0 to 1711  
Data columns (total 9 columns):  
 #   Column             Non-Null Count  Dtype  
---  ---  
 0   id_order            1712 non-null   int64  
 1   id_produk           1712 non-null   int64  
 2   nama_produk         1712 non-null   object  
 3   kategori_produk     1712 non-null   category  
 4   tanggal_pembelian  1712 non-null   datetime64[ns]  
 5   kuantitas           1712 non-null   int64  
 6   total_harga         1712 non-null   int64  
 7   PPN                 1712 non-null   float64  
 8   metode_bayar        1712 non-null   object  
dtypes: category(1), datetime64[ns](1), float64(1), int64(4), object(2)  
memory usage: 189.0+ KB
```

***Pada solusi, digunakan google colab.**



Thank You!



KALBE
Nutritional