



Pelatihan Microcredential CERTIFICATION untuk Associate Data Scientist

1 November - 10 Desember 2021



Hands-On

Hands-On ini digunakan pada kegiatan Microcredential Associate Data Scientist 2021

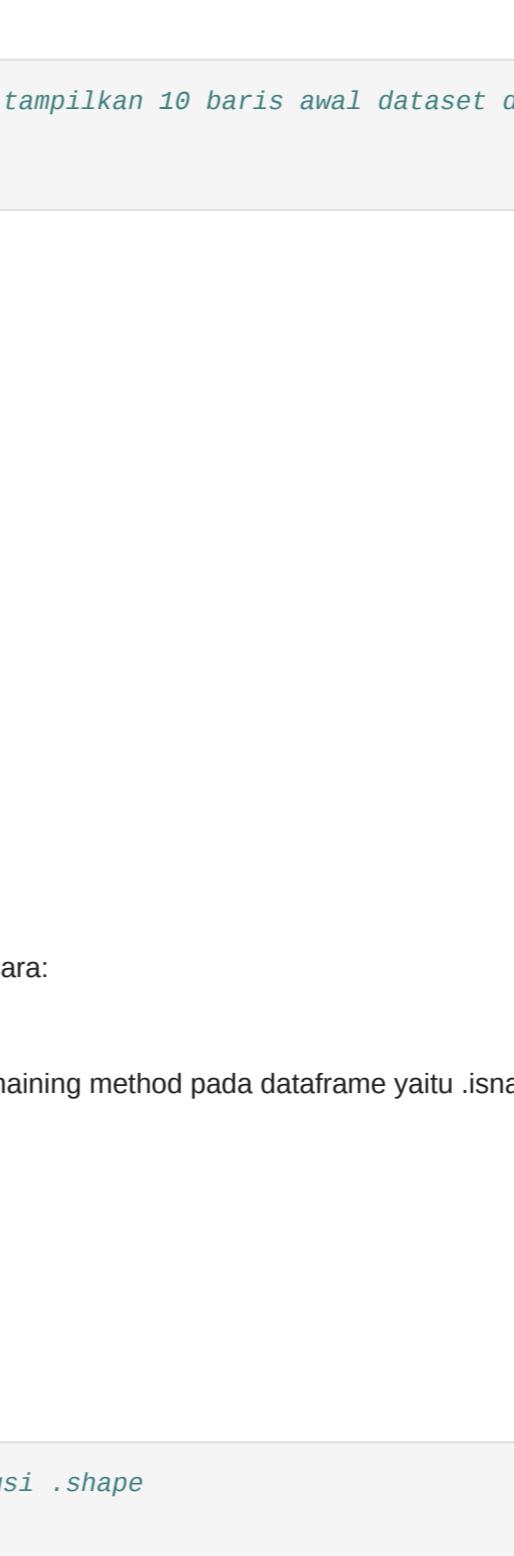
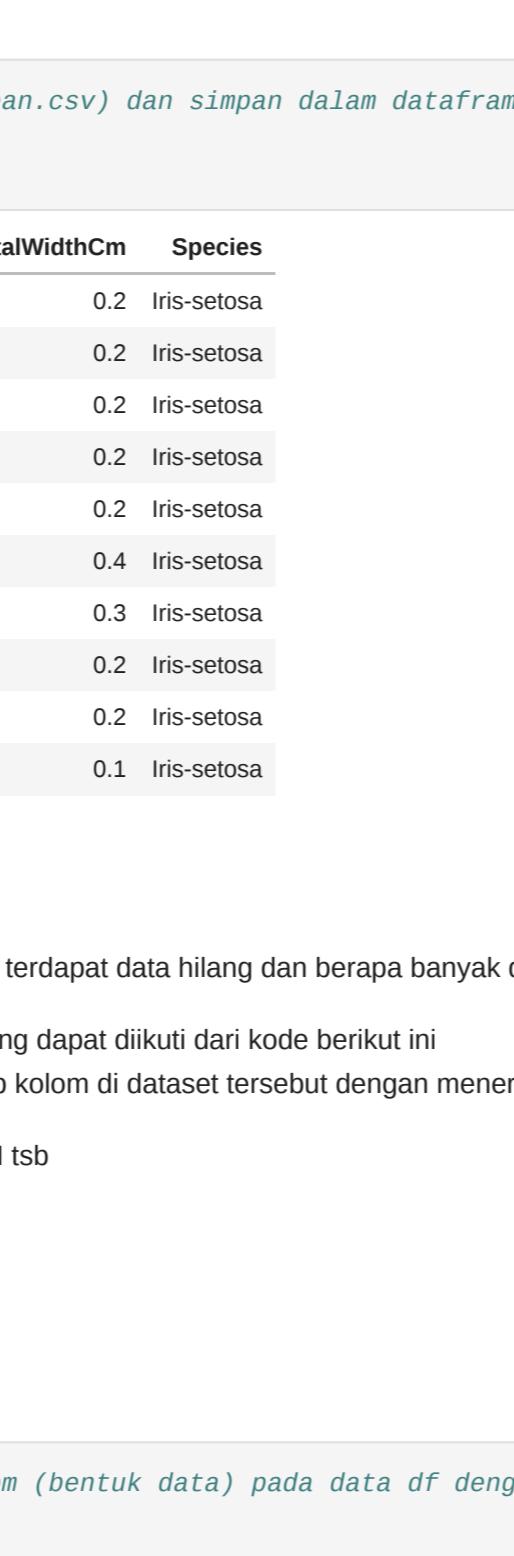
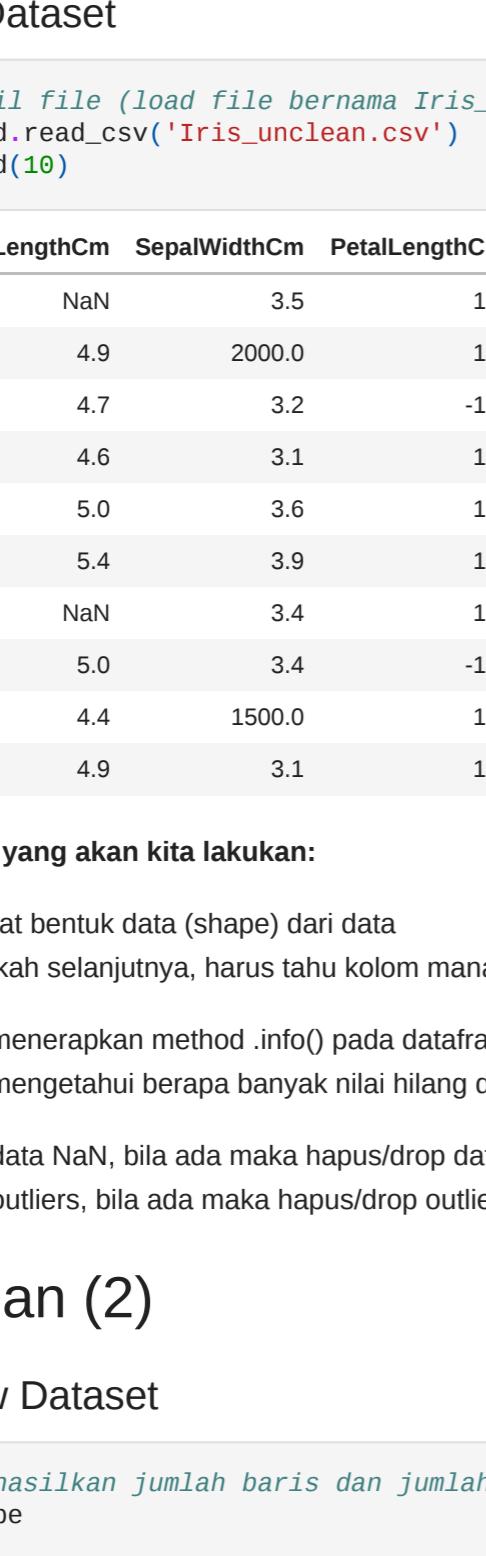
Pertemuan 8

Pertemuan 8 (delapan) pada Microcredential Associate Data Scientist 2021 menyampaikan materi mengenai Membersihkan Data dan Memvalidasi Data

DATA CLEANSING & Handling Missing Values

Value yang hilang serta tidak lengkap dari dataframe akan membuat analisis atau model prediksi yang dibuat menjadi tidak akurat dan mengakibatkan keputusan salah yang diambil. Terdapat beberapa cara untuk mengatasi data yang hilang/tidak lengkap tersebut.

Kali ini, kita akan menggunakan Dataset Iris yang kotor / terdapat nilai NaN dan outliers



Info dataset: Dataset ini berisi ukuran/measures 3 spesies iris

Pada Tugas Mandiri Pertemuan 8

silakan Anda kerjakan Latihan 1 s/d 20. Output yang anda lihat merupakan panduan yang dapat Anda ikuti dalam penulisan code :)

Latihan (1)

Melakukan import library yang dibutuhkan

```
In [2]: # import library pandas
import pandas as pd

# import library numpy
import numpy as np

# import library matplotlib
import matplotlib.pyplot as plt

# import library seaborn
import seaborn as sns

# me non aktifkan peringatan pada python dengan import warning -> 'ignore'
import warnings
warnings.filterwarnings("ignore")
```

Load Dataset

```
In [3]: #Panggil file (load file bernama Iris_unclean.csv) dan simpan dalam datafarme Lalu tampilkan 10 baris awal dataset dengan function head()
df = pd.read_csv('Iris_unclean.csv')
df.head(10)
```

```
Out[3]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
0   NaN        3.5       1.4      0.2  Iris-setosa
1   4.9        2.0000    1.4      0.2  Iris-setosa
2   4.7        3.2       -1.3     0.2  Iris-setosa
3   4.6        3.1       1.5      0.2  Iris-setosa
4   5.0        3.6       1.4      0.2  Iris-setosa
5   5.4        3.9       1.7      0.4  Iris-setosa
6   NaN        3.4       1.4      0.3  Iris-setosa
7   5.0        3.4       -1.5     0.2  Iris-setosa
8   4.4        1500.0    1.4      0.2  Iris-setosa
9   4.9        3.1       1.5      0.1  Iris-setosa
```

Kegiatan yang akan kita lakukan:

- Melihat bentuk data (shape) dari data
- Langkah selanjutnya, harus tahu kolom mana yang terdapat data hilang dan berapa banyak dengan cara:
 - menerapkan method .info() pada dataframe yang dilakukan di kode berikut ini
 - mengetahui berapa banyak nilai hilang di tiap kolom di dataset tersebut dengan menerapkan chaining method pada datafarme yaitu .isna().sum()
- Cek data NaN, bila ada maka hapus/drop data NaN tsb
- Cek outliers, bila ada maka hapus/drop outliers tsb

Latihan (2)

Review Dataset

```
In [4]: # menghasilkan jumlah baris dan jumlah kolom (bentuk data) pada data df dengan fungsi .shape
df.shape
```

```
Out[4]: (150, 5)
```

```
In [6]: # fungsii describe() untuk mengetahui statistika data untuk data numeric seperti count, mean, standard deviation, maximum, minimum, dan quartile.
df.describe()
```

```
Out[6]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm
count    148.000000  150.000000  150.000000  150.000000
mean     5.856757  26.348000  3.721333  1.198667
std      0.824964  203.117929  1.842364  0.763161
min      4.300000  2.000000  -1.500000  0.100000
25%     5.100000  2.800000  1.600000  0.300000
50%     5.800000  3.000000  4.350000  1.300000
75%     6.400000  3.375000  5.100000  1.800000
max     7.900000  2000.000000  6.900000  2.500000
```

```
In [7]: # Informasi lebih detail mengenai struktur DataFrame dapat dilihat menggunakan fungsi info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype  
 --- 
 0   SepalLengthCm   140 non-null   float64
 1   SepalWidthCm   140 non-null   float64
 2   PetalLengthCm  140 non-null   float64
 3   PetalWidthCm  140 non-null   float64
 4   Species       140 non-null   object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
In [9]: #cek nilai yang hilang / missing values di dalam data
df.isna().sum()
```

```
Out[9]: SepalLengthCm    2
SepalWidthCm    0
PetalLengthCm   0
PetalWidthCm   0
Species        0
dtype: int64
```

Missing values adalah nilai yang tidak terdefinisi di dataset. Bentuknya beragam, bisa berupa blank cell, ataupun simbol-simbol tertentu seperti NaN (Not a Number), NA (Not Available), ?, ., dan sebagainya. Missing values dapat menjadi masalah dalam analisis data serta tentunya dapat mempengaruhi hasil modelling machine learning. Dari hasil diatas dataset tsb mengandung 2 data missing values pada kolom 'SepalLengthCm' dan beberapa outliers!

Periksa dan Cleansing setiap kolom pada data

dalam kasus ini hint nya adalah: hanya kolomfield 'SepalLengthCm' 'SepalWidthCm' 'PetalLengthCm' yang bermasalah dan kita hanya akan berfokus cleansing pada kolom/field tsb

1. Kolom SepalLengthCm

Latihan (3)

periksa statistik data kolom SepalLengthCm

```
In [11]: df['SepalLengthCm'].describe()
```

```
Out[11]: count    148.000000
mean     5.856757
std      0.824964
min      4.300000
25%     5.100000
50%     5.800000
75%     6.400000
max     7.900000
Name: SepalLengthCm, dtype: float64
```

Dari data diatas terlihat pada terdapat kejanggalan pada nilai max yaitu bernilai minus, sedangkan Petal Length/ lebar Kelopak bunga nampaknya tidak masuk akal bila berukuran hingga 2000cm.

Sehingga dapat dipastikan ini merupakan outliers

Latihan (4)

periksa jumlah nilai NaN pada kolom SepalLengthCm

```
In [12]: print('Nilai NaN pada kolom SepalLengthCm berjumlah : ', df['SepalLengthCm'].isna().sum())
```

```
Nilai NaN pada kolom SepalLengthCm berjumlah : 2
```

Latihan (5)

cetak index dari nilai NaN kolom SepalLengthCm dengan function np.where

```
In [13]: index_nan = np.where(df['SepalLengthCm'].isna())
```

```
Out[13]: (array([0, 6], dtype=int64),)
```

Latihan (6)

1. Cetak ukuran/dimensi dari datafarme

2. Drop baris jika ada satu saja yang missing dengan function dropna() dan cetak ukurannya

```
In [14]: df = df.dropna()
print("Ukuran awal df: %d baris, %d kolom." % df.shape)
```

Ukuran awal df: 150 baris, 5 kolom.

Ukuran df setelah dibuang baris yang memiliki missing value: %d baris, %d kolom." % df.shape)

2. Kolom SepalWidthCm

Latihan (7)

periksa statistik data kolom SepalWidthCm

```
In [15]: df['SepalWidthCm'].describe()
```

```
Out[15]: count    148.000000
mean     26.657432
std      204.477337
min      2.000000
25%     2.800000
50%     3.000000
75%     3.300000
max     2000.000000
Name: SepalWidthCm, dtype: float64
```

Dan data diatas terlihat pada terdapat kejanggalan pada nilai min yaitu bernilai minus, sedangkan Petal Length/ lebar Kelopak bunga nampaknya tidak masuk akal bila berukuran hingga 2000cm.

Sehingga dapat dipastikan ini merupakan outliers

Latihan (8)

mendeteksi outlier dengan menggunakan boxplot pada kolom SepalWidthCm

```
In [20]: plt.plot(figsize=(10, 5))
sns.boxplot(df['SepalWidthCm'])
plt.annotate('Outlier', (df['SepalWidthCm'].describe()['max']+0.1, 1.7), xytext=(df['SepalWidthCm'].describe()['max'], 0.3), arrowprops=dict(facecolor='blue', fontsize=13))
IQR = df['SepalWidthCm'].describe()['75%'] - df['SepalWidthCm'].describe()['25%']
```


Latihan (9)

membuat fungsi melihat data outlier dengan rumus IQR = Q3-Q1

```
In [21]: def detect_outliers(df, x):
    Q1 = df[x].describe()['25%']
    Q3 = df[x].describe()['75%']
    IQR = Q3-Q1
    return df[(df[x] < Q1-1.5*IQR) | (df[x] > Q3+1.5*IQR)]
```

Latihan (10)

melihat data outliers dari kolom SepalWidthCm menggunakan fungsi yang telah dibuat

```
In [26]: detect_outliers(df, 'SepalWidthCm')
```

```
Out[26]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	4.9	2.0000	1.4	0.2	Iris-setosa
8	4.4	1500.0	1.4	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa
32	5.2	4.1	1.5	0.1	Iris-setosa
33	5.5	4.2	1.4	0.2	Iris-setosa
60	5.0	2.0	3.5	1.0	Iris-versicolor

Latihan (11)

hapus data outlier dari kolom SepalWidthCm

```
In [28]: df = df.drop((df[df['SepalWidthCm']>4]).index, axis=0)
```

```
Out[28]: df = df.drop((df[df['SepalWidthCm']<2.1]).index, axis=0)
```

Latihan (12)

cek ulang outliers dengan fungsi yang telah dibuat

```
In [31]: detect_outliers(df, 'SepalWidthCm')
```

```
Out[31]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
2	4.7	3.2	-1.3	0.2	Iris-setosa
7	5.0	3.4	-1.5	0.2	Iris-setosa

Latihan (13)

cek ulang outliers dengan boxplot pada kolom SepalWidthCm

```
In [32]: plt.plot(figsize=(10, 5))
sns.boxplot(df['SepalWidthCm'])
```

```
<AxesSubplot:xlabel='SepalWidthCm'>
```


Latihan (14)

periksa statistik data kolom PetalLengthCm

```
In [34]: df['PetalLengthCm'].describe()
```

```
Out[34]: count    148.000000
mean     3.825915
std      1.819988
min      -1.500000
25%     2.800000
50%     3.000000
75%     3.300000
max     6.900000
Name: PetalLengthCm, dtype: float64
```

Dari data diatas terlihat pada terdapat kejanggalan pada nilai min yaitu bernilai minus, sedangkan Petal Length/ lebar Kelopak bunga nampaknya tidak masuk akal bila berukuran minus. Sehingga

dapat dipastikan ini merupakan outliers

Latihan (15)

hapus data bernilai minus / outlier kolom PetalLengthCm

```
In [42]: df = df.drop((df[df['PetalLengthCm']<0]).index, axis=0)
```

Latihan (16)

cek ulang outliers dengan fungsi yang telah dibuat

```
In [43]: df[df['PetalLengthCm']<1]
```

```
Out[43]: SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
2	4.7	3.2	-1.3	0.2	Iris-setosa
7	5.0	3.4	-1.5	0.2	Iris-setosa

CEK DATA SETELAH PROSES CLEANSING</h