



Pelatihan
Microcredential
CERTIFICATION untuk
Associate Data Scientist
1 November - 10 Desember 2021

Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Republik Indonesia

IARC
DIKTI
SIGAP MELAYANI
Kampus Merdeka INDONESIA-JAYA

Hands-On

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

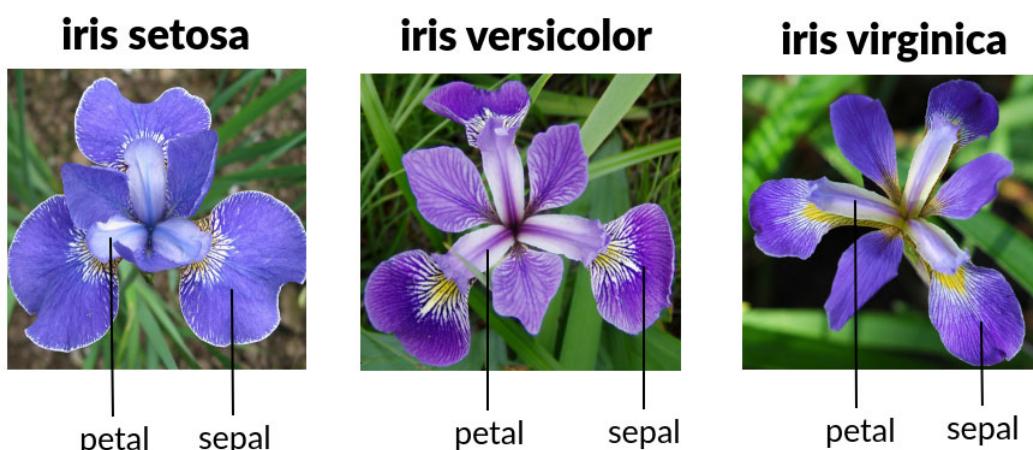
Pertemuan 8

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

Latihan (1)

Melakukan import library yang dibutuhkan

In [1]:

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

```
#Panggil file (load file bernama Iris_unclean.csv) dan simpan dalam dataframe
df = pd.read_csv('Iris_unclean.csv')
df.head(10)
```

Out [2]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	NaN	3.5	1.4	0.2	Iris-setosa
1	4.9	2000.0	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:

1. menerapkan method .info() pada dataframe yang dapat diikuti dari kode berikut ini
2. mengetahui berapa banyak nilai hilang dari tiap kolom di dataset tersebut dengan menerapkan chaining method pada dataframe 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 [3]: # menghasilkan jumlah baris dan jumlah kolom (bentuk data) pada data df dengan df.shape
```

```
Out[3]: (150, 5)
```

```
In [4]: # fungsi describe() untuk mengetahui statistika data untuk data numeric seperti df.describe()
```

	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 [5]: # Informasi lebih detail mengenai struktur DataFrame dapat dilihat menggunakan 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    148 non-null    float64 
 1   SepalWidthCm     150 non-null    float64 
 2   PetalLengthCm    150 non-null    float64 
 3   PetalWidthCm     150 non-null    float64 
 4   Species          150 non-null    object  
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [6]:

```
#cek nilai yang hilang / missing values di dalam data  
df.isnull().sum()
```

```
Out[6]: 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/field 'SepalLengthCm' dan beberapa outliers!

Periksa dan Cleansing setiap kolom pada data

dalam kasus ini hint nya adalah: hanya kolom/field '**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 [7]:

```
df['SepalLengthCm'].describe()
```

```
Out[7]: 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
```

Latihan (4)

periksa jumlah nilai NaN pada kolom SepalLengthCm

In [8]:

```
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 [9]: index_nan = np.where(df['SepalLengthCm'].isnull())
index_nan
```

```
Out[9]: (array([0, 6], dtype=int64),)
```

```
In [ ]:
```

Latihan (6)

1. Cetak ukuran/dimensi dari dataframe
2. Drop baris jika ada satu saja data yang missing dan ukuran/dimensi dari dataframe setelah di drop

```
In [10]: # Cetak ukuran awal dataframe
print("Ukuran awal df: %d baris, %d kolom." % (len(df.index), len(df.columns)))

# Drop baris jika ada satu saja data yang missing dengan function dropna() dari df
df = df.dropna()
print("Ukuran df setelah dibuang baris yang memiliki missing value: %d baris,"
```

Ukuran awal df: 150 baris, 5 kolom.

Ukuran df setelah dibuang baris yang memiliki missing value: 148 baris, 5 kolom.

2. Kolom SepalWidthCm

Latihan (7)

periksa statistik data kolom SepalWidthCm

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

```
Out[11]:      SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count        148.000000    148.000000    148.000000    148.000000
mean         5.856757     26.657432     3.752703     1.211486
std          0.824964    204.477337     1.834716     0.760194
min          4.300000     2.000000    -1.500000     0.100000
25%          5.100000     2.800000     1.600000     0.300000
50%          5.800000     3.000000     4.400000     1.300000
75%          6.400000     3.300000     5.100000     1.800000
```

SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

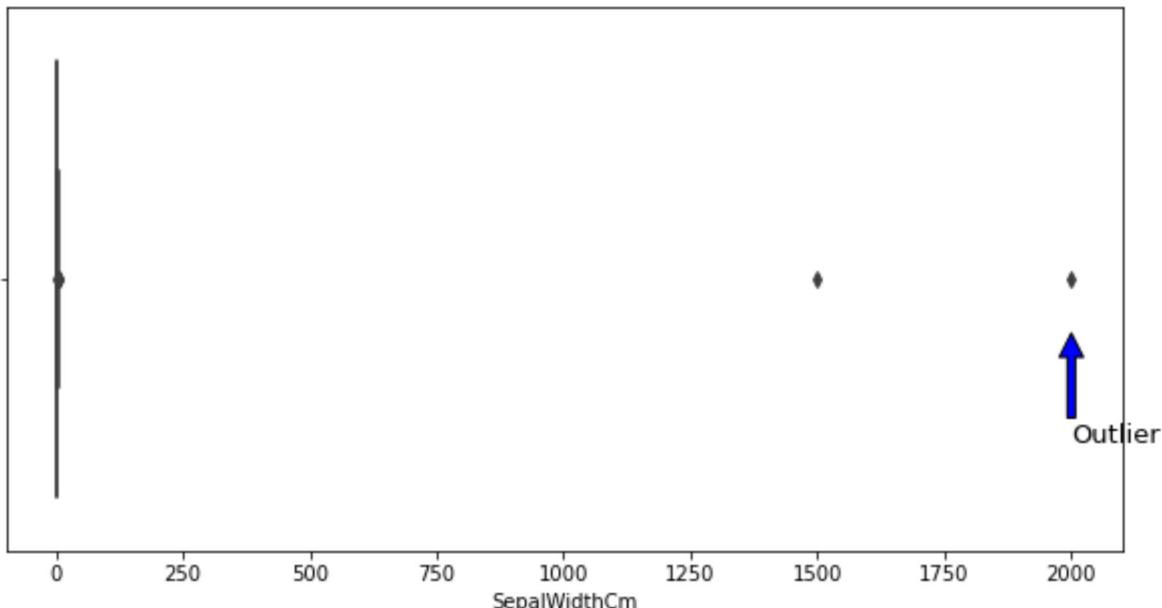
Dari data diatas terlihat pada terdapat kejanggalan pada nilai max yaitu 2000cm, sedangkan Sepal Width/ lebar Kelopak bunga nampaknya tidak masuk akal bila berukuran hingga 2000cm. Sehingga dapat dipastikan ini merupakan outliers

Latihan (8)

mendeteksi outlier dengan menggunakan boxplot pada kolom SepalWidthCm

In [12]:

```
plt.figure(figsize = (10, 5))
sns.boxplot(df['SepalWidthCm'])
plt.annotate('Outlier', (df['SepalWidthCm'].describe()['max'], 0.1), xytext =
            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 [13]:

```
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)

```
In [14]: detect_outliers(df, 'SepalWidthCm')
```

```
Out[14]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	4.9	2000.0	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 [15]: df = df.drop((df[df['SepalWidthCm'] > 4]).index, axis=0)
```

```
In [16]: df = df.drop((df[df['SepalWidthCm'] < 2.1]).index, axis=0)
```

Latihan (12)

cek ulang outliers dengan fungsi yang telah dibuat

```
In [17]: detect_outliers(df, 'SepalWidthCm')
```

```
Out[17]:
```

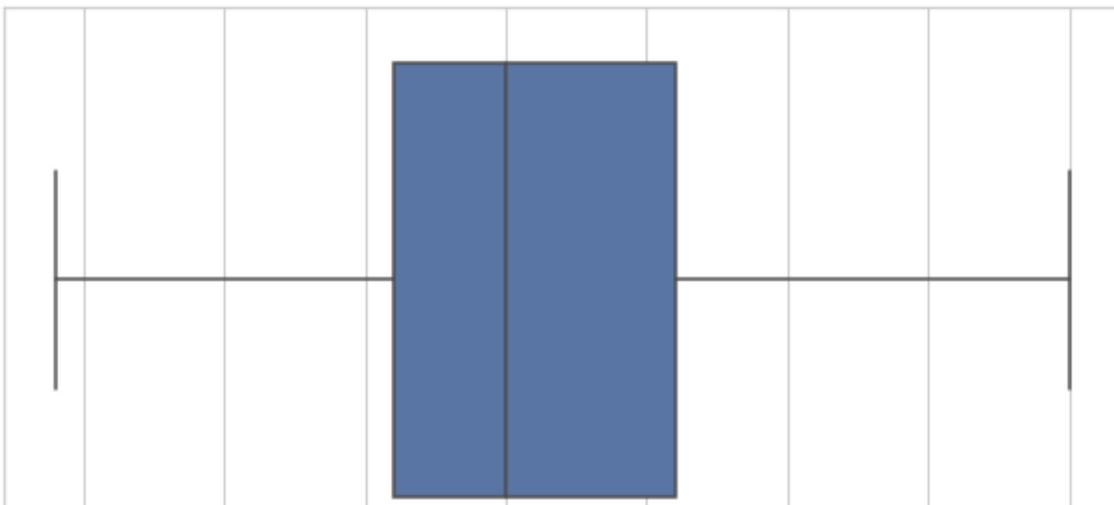
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
--	---------------	--------------	---------------	--------------	---------

Latihan (13)

cek ulang outliers dengan boxplot

```
In [18]: plt.figure(figsize = (10, 5))
sns.set_theme(style="whitegrid")
sns.boxplot(df['SepalWidthCm'])
```

```
Out[18]: <AxesSubplot:xlabel='SepalWidthCm'>
```



3. Kolom PetalLengthCm

Latihan (14)

periksa statistik data kolom SepalLengthCm

```
In [19]: df['SepalLengthCm'].describe()
```

```
Out[19]: count    142.000000
mean      5.888028
std       0.823199
min       4.300000
25%      5.125000
50%      5.800000
75%      6.400000
max      7.900000
Name: SepalLengthCm, dtype: float64
```

Dari data diatas terlihat pada terdapat kejanggalan pada nilai min yaitu bernilai minus, sedangkan Petal Length/ panjang Kelopak bunga nampaknya tidak masuk akal bila berukuran minus. Sehingga dapat dipastikan ini merupakan outliers

Latihan (15)

periksa data bernilai minus pada kolom PetalLengthCm

```
In [20]: df.loc[df['PetalLengthCm'] < 1]
```

	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 (16)

hapus data bernilai minus / outlier kolom PetalLengthCm

```
In [21]: df = df.drop((df[df['PetalLengthCm'] < 1]).index, axis=0)
```

Latihan (17)

cek ulang outliers dengan fungsi yang telah dibuat

```
In [22]: df.loc[df['PetalLengthCm'] < 1]
```

```
Out[22]: SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  Species
```

CEK DATA SETELAH PROSES CLEANSING

Latihan (18)

Melihat nomor index beserta tipe datanya dengan function info()

```
In [23]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 140 entries, 3 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: 6.6+ KB
```

Latihan (19)

cek ulang nilai yang hilang / missing values di dalam data setelah proses cleansing

```
In [24]: df.isnull().sum()
```

```
Out[24]: SepalLengthCm      0
SepalWidthCm       0
PetalLengthCm      0
PetalWidthCm       0
```

Species 0

Latihan (20)

Tampilkan 10 baris dataframe setelah proses cleansing

In [27]: df.head(10)

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa

DATA SUDAH SIAP UNTUK KETAHAP SELANJUTNYA YAITU MODELLING :)