



## Hands-On

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

## Pertemuan 9

Pertemuan 9 (sembilan) pada Microcredential Associate Data Scientist 2021 menyampaikan materi mengenai Mengkonstruksi Data

### Pada Tugas Mandiri Pertemuan 9

silakan Anda kerjakan Latihan 1 s/d 10. Output yang anda lihat merupakan panduan yang dapat Anda ikuti dalam penulisan code :)

#### Latihan (1)

##### Melakukan import library yang dibutuhkan

```
In [1]: # import Library pandas
import pandas as pd
# Import Library scipy
import scipy
# Import Library winsorize dari scipy
from scipy.stats.mstats import winsorize
# Import Library trima dari scipy
from scipy.stats.mstats import trim
# Import Library RandomSampleImputer dari feature engine imputation
from feature_engine.imputation import RandomSampleImputer
# import Library StandardScaler dari sklearn
from sklearn.preprocessing import StandardScaler
```

## Latihan (2)

Menghitung nilai null pada dataset :

1. Load dataset Iris\_Unclean
2. Tampilkan dataset
3. Hitung jumlah nilai null pada dataset

In [2]:

```
# Load dataset Iris_Unclean  
df = pd.read_csv('Iris_unclean.csv')
```

In [3]:

```
# tampilkan dataset  
df
```

Out[3]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [5]:

```
# hitung jumlah nilai null pada dataset  
df.isna().sum()
```

Out[5]:

```
SepalLengthCm    2  
SepalWidthCm    0  
PetalLengthCm   0  
PetalWidthCm    0  
Species         0  
dtype: int64
```

## Latihan (3)

Melakukan handle missing value dengan Imputasi Mean:

1. Load dataset Iris\_Unclean
2. Ambil 10 data teratas "SepalLengthCm", kemudian tampilkan
3. Mengganti missing value Imputasi dengan mean, kemudian masukkan ke variable
4. Tampilkan 10 data teratas "SepalLengthCm" setelah handle missing value dengan Imputasi mean()

```
In [10]: # Load dataset Iris_Unclean
df = pd.read_csv('Iris_unclean.csv')
```

```
In [11]: # ambil 10 data teratas SepalLengthCm, kemudian tampilkan
df = df.loc[:9, 'SepalLengthCm']
df
```

```
Out[11]: 0    NaN
          1    4.9
          2    4.7
          3    4.6
          4    5.0
          5    5.4
          6    NaN
          7    5.0
          8    4.4
          9    4.9
Name: SepalLengthCm, dtype: float64
```

```
In [12]: # mengganti missing value dengan mean(), kemudian masukkan ke variabel
df = df.fillna(df.mean())
```

```
In [13]: # tampilkan 10 data teratas SepalLengthCm setelah handle missing value dengan imputasi
df
```

```
Out[13]: 0    4.8625
          1    4.9000
          2    4.7000
          3    4.6000
          4    5.0000
          5    5.4000
          6    4.8625
          7    5.0000
          8    4.4000
          9    4.9000
Name: SepalLengthCm, dtype: float64
```

## Latihan (4)

**Melakukan handle missing value dengan nilai suka-suka (Arbitrary):**

1. Load dataset Iris\_Unclean
2. Ambil 10 data teratas "SepalLengthCm", kemudian tampilkan
3. Mengganti missing value dengan imputasi nilai suka-suka (Arbitrary), kemudian masukkan ke variable

4. Tampilkan 10 data teratas "SepalLengthCm" setelah handle missing value dengan nilai suka-suka

In [14]:

```
# Load dataset Iris_Unclean  
df = pd.read_csv('Iris_unclean.csv')
```

In [15]:

```
# ambil 10 data teratas SepalLengthCm, kemudian tampilkan  
df = df.loc[:9, 'SepalLengthCm']  
df
```

Out[15]:

```
0    NaN  
1    4.9  
2    4.7  
3    4.6  
4    5.0  
5    5.4  
6    NaN  
7    5.0  
8    4.4  
9    4.9  
Name: SepalLengthCm, dtype: float64
```

In [16]:

```
# melakukan imputasi nilai suka-suka (Arbitrary), masukkan ke dalam variabel  
df = df.fillna(99)
```

In [17]:

```
# tampilkan 10 data teratas SepalLengthCm setelah handle missing value dengan nilai  
df
```

Out[17]:

```
0    99.0  
1    4.9  
2    4.7  
3    4.6  
4    5.0  
5    5.4  
6    99.0  
7    5.0  
8    4.4  
9    4.9  
Name: SepalLengthCm, dtype: float64
```

## Latihan (5)

Melakukan handle missing value dengan frequent category / modus:

1. Load dataset Iris\_Unclean
2. Ambil 10 data teratas "SepalLengthCm", kemudian tampilkan
3. Mengganti missing value dengan frequent category / modus
4. Tampilkan hasil imputasi "SepalLengthCm" setelah handle dengan frequent category / modus

In [40]:

```
# Load dataset Iris_Unclean
```

```
df = pd.read_csv('Iris_unclean.csv')
```

```
In [41]: # tampilkan 10 data teratas kolom SepalLengthCm  
df = df.loc[:9, 'SepalLengthCm'].to_frame()  
df
```

```
Out[41]: SepalLengthCm
```

	SepalLengthCm
0	NaN
1	4.9
2	4.7
3	4.6
4	5.0
5	5.4
6	NaN
7	5.0
8	4.4
9	4.9

```
In [42]: # Import SimpleImputer dari sklearn.impute  
  
from sklearn.impute import SimpleImputer  
  
# Mengatasi missing value dengan frequent category / modus  
  
imp = SimpleImputer(strategy='most_frequent')
```

```
In [43]: # Tampilkan hasil imputasi "SepalLengthCm"  
  
imp.fit_transform(df)
```

```
Out[43]: array([[4.9],  
                 [4.9],  
                 [4.7],  
                 [4.6],  
                 [5. ],  
                 [5.4],  
                 [4.9],  
                 [5. ],  
                 [4.4],  
                 [4.9]])
```

## Latihan (6)

Melakukan handle missing value dengan Imputasi Random Sample:

1. Load dataset Iris\_Unclean
2. Tampilkan 10 data teratas

3. Membuat imputer random sample dengan random state = 5
4. Cocokan imputer ke data
5. Ubah data dengan imputer masukkan ke dalam variable
6. Tampilkan hasil imputasi data "SepalLengthCm"

In [55]:

```
# Load dataset Iris_Unclean  
  
df = pd.read_csv('Iris_unclean.csv')
```

In [56]:

```
# tampilkan 10 data teratas SepalLengthCm  
  
df = df.loc[:9, 'SepalLengthCm'].to_frame()  
df
```

Out[56]:

	SepalLengthCm
0	NaN
1	4.9
2	4.7
3	4.6
4	5.0
5	5.4
6	NaN
7	5.0
8	4.4
9	4.9

In [57]:

```
# Membuat imputer random sample dengan random state = 5  
  
imputer = RandomSampleImputer(random_state=5)  
  
# Cocokan imputer ke data  
  
imputer.fit(df)  
  
# Ubah data dengan imputer masukkan ke dalam variable  
  
test_t = imputer.transform(df)
```

In [58]:

```
# Tampilkan data hasil imputasi data "SepalLengthCm"  
  
test_t
```

Out[58]:

	SepalLengthCm
0	5.0
1	4.9
2	4.7
3	4.6

	SepalLengthCm
4	5.0
5	5.4
6	4.4
7	5.0
8	4.4
9	4.9

## Latihan (7)

### Melakukan Winsorizing

1. Import library winsorize dari scipy
2. Load data Iris\_AfterClean
3. Ambil 10 data teratas "SepalLengthCm", kemudian masukkan ke dalam variabel datan tampilkan
4. Winsorize data dengan batas nilai terendah 10% dan batas nilai tinggi 20%
5. Tampilkan hasil winsorize

In [59]:

```
# Import library scipy
import scipy
```

In [60]:

```
# Load data Iris_AfterClean
data = pd.read_csv('Iris_AfterClean.csv')

# Ambil 10 data teratas "SepalLengthCm", kemudian masukkan ke dalam variabel datan t
a = data.loc[:9, 'SepalLengthCm']
a
```

Out[60]:

```
0    4.6
1    5.0
2    5.4
3    4.9
4    5.4
5    4.8
6    4.8
7    4.3
8    5.8
9    5.4
Name: SepalLengthCm, dtype: float64
```

In [61]:

```
# Winsorize data dengan batas nilai terendah 10% dan batas nilai tinggi 20%
wins = winsorize(a, limits=[0.1, 0.2])

# Tampilkan hasil winsorize
print(wins)
```

[4.6 5. 5.4 4.9 5.4 4.8 4.8 4.6 5.4 5.4]

# Latihan (8)

## Melakukan Trimming

1. Import library trima dari scipy
2. Load data Iris\_AfterClean
3. Ambil 10 data teratas "SepalLengthCm", kemudian masukkan ke dalam variabel datan tampilkan
4. Trimming data dengan batas nilai terendah 2 dan batas nilai tinggi 5
5. Tampilkan hasil trimming

In [62]:

```
# Import Library trima dari scipy
from scipy.stats.mstats import trim
```

In [63]:

```
# Load data Iris_AfterClean
data = pd.read_csv('Iris_AfterClean.csv')

# Ambil 10 data teratas "SepalLengthCm", kemudian masukkan ke dalam variabel datan t
a = data.loc[:9, 'SepalLengthCm']
a
```

Out[63]:

```
0    4.6
1    5.0
2    5.4
3    4.9
4    5.4
5    4.8
6    4.8
7    4.3
8    5.8
9    5.4
Name: SepalLengthCm, dtype: float64
```

In [65]:

```
# Trimming data dengan batas nilai terendah 2 dan batas nilai tinggi 5
trims = trim(a, limits=(2,5))

# Tampilkan hasil trimming
print(trim)
```

```
[4.6 5.0 -- 4.9 -- 4.8 4.8 4.3 -- --]
```

# Latihan (9)

## Melakukan Scaling: Normalisasi

1. Load data Iris\_AfterClean
2. Ambil 10 data teratas SepalLengthCm dan SepalWidthCm
3. Menghitung mean data
4. Menghitung max - min pada data
5. Menerapkan transformasi ke data
6. Tampilkan hasil scalling

```
In [66]: # Load data Iris_AfterClean
data = pd.read_csv('Iris_AfterClean.csv')

# Ambil 10 data teratas SepalLengthCm dan SepalWidthCm

data = data.loc[:9, ['SepalLengthCm', 'SepalWidthCm']]
data
```

Out[66]:

	SepalLengthCm	SepalWidthCm
0	4.6	3.1
1	5.0	3.6
2	5.4	3.9
3	4.9	3.1
4	5.4	3.7
5	4.8	3.4
6	4.8	3.0
7	4.3	3.0
8	5.8	4.0
9	5.4	3.9

```
In [67]: # Menghitung mean
means = data.mean(axis=0)

# menghitung max - min
max_min = data.max(axis=0) - data.min(axis=0)

# menerapkan transformasi ke data
train_scaled = (data - means) / max_min
```

```
In [68]: # Tampilkan hasil scalling
train_scaled
```

Out[68]:

	SepalLengthCm	SepalWidthCm
0	-0.293333	-0.37
1	-0.026667	0.13
2	0.240000	0.43
3	-0.093333	-0.37
4	0.240000	0.23
5	-0.160000	-0.07
6	-0.160000	-0.47
7	-0.493333	-0.47
8	0.506667	0.53
9	0.240000	0.43

# Latihan (10)

## Melakukan Scaling: Standardisasi

1. Load data Iris\_AfterClean
2. Ambil 10 data teratas SepalLengthCm dan SepalWidthCm
2. Import library StandardScaler dari sklearn
3. Membuat objek scaler
4. Sesuaikan scaler dengan data
5. Mengubah data
6. Tampilkan hasil scaling dengan standarisasi

In [69]:

```
# Load data Iris_AfterClean
data = pd.read_csv('Iris_AfterClean.csv')

# Ambil 10 data teratas SepalLengthCm dan SepalWidthCm
data = data.loc[:9, ['SepalLengthCm', 'SepalWidthCm']]
data
```

Out[69]:

	SepalLengthCm	SepalWidthCm
0	4.6	3.1
1	5.0	3.6
2	5.4	3.9
3	4.9	3.1
4	5.4	3.7
5	4.8	3.4
6	4.8	3.0
7	4.3	3.0
8	5.8	4.0
9	5.4	3.9

In [70]:

```
# import Library StandardScaler dari sklearn
from sklearn.preprocessing import StandardScaler

# Buat objek scaler
scaler = StandardScaler()

# Sesuaikan scaler dengan data
scaler.fit(data)

# Mengubah data
train_scaled = scaler.transform(data)
```

In [71]:

```
# Tampilkan hasil
train_scaled
```

```
Out[71]: array([[-1.02464215, -0.97469723],  
   [-0.09314929,  0.34246119],  
   [ 0.83834358,  1.13275625],  
   [-0.3260225 , -0.97469723],  
   [ 0.83834358,  0.60589288],  
   [-0.55889572, -0.18440218],  
   [-0.55889572, -1.23812892],  
   [-1.7232618 , -1.23812892],  
   [ 1.76983644,  1.39618793],  
   [ 0.83834358,  1.13275625]])
```