Laporan Praktikum Pertemuan 2

Data Science Lanjut

Dibuat Oleh

Nama	: Muhamad Faisal Halim
NIM	: 19.240.0163
Kelas	: -
Mata Kuliah	: Data Science Lanjut

Mahasiswa Pertukaran Mahasiswa.

Universitas Muhammadiyah Kalimantan Timur

~ STMIK Widya Pratama Pekalongan

Note

Data pada praktikum ini disamakan dengan data yang ada pada contoh yang diberikan di Openlearning UMKT.

Praktikum

Membaca File Pada Python Menggunakan Pandas

pada praktikum ini, kita akan mencoba membaca file dengan pandas pada python. Pandas adalah salah satu library untuk bisa membaca file dan proses awal dari analisis data. file yang dapat di baca oleh pandas antara lain. csv, txt, tsv dll.

```python
import pandas as pd

csv_data = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print(csv_data)
```

Python

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male   19                  15                      39
1             2    Male   21                  15                      81
2             3  Female   20                  16                       6
3             4  Female   23                  16                      77
4             5  Female   31                  17                      40
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[200 rows x 5 columns]
```

masih menggunakan pandas. terkadang kita memiliki data yang jumlahnya luarbiasa banyak, hal itu akan menyebabkan proses dan loading yang lama. jadi untuk memastikan data terbaca dengan aman, kita bisa menggunakan fungsi .`head()` dari pandas. secara default fungsi `.head()` pada pandas akan menampilkan 5 data dari jumlah kesluruhan data yang dibaca.

```python
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print(csv.head(7))
```

✓ 0.5s Python

```
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
5           6  Female   22                  17                      76
6           7  Female   35                  18                       6
```

kita juga dappat menampilkan hanya 1 kolom dari table file csv kita, untuk contohnya seperti dibawah ini.

```python
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print(csv['Age'].head(10))
```

✓ 0.6s Python

```
0    19
1    21
2    20
3    23
4    31
5    22
6    35
7    23
8    64
9    30
Name: Age, dtype: int64
```

selain itu kita juga dapat mengkombinasikanya dengan `.head()` untuk hanya menampilkan sebagian data saja. dan selain dapat menampilkan hanya 1 kolom saja, panda juga bisa membantu kita mengintip 1 data dari baris tertentu.

```python
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print(csv.iloc[3])
```

✓ 5.8s Python

```
CustomerID                   4
Genre                   Female
Age                         23
Annual Income (k$)          16
Spending Score (1-100)      77
Name: 3, dtype: object
```

selain itu, kita juha bisa menkombinasikan baris dan kolomnya, jadi kita bisa mengambil 1 baris data dari kolom tertentu. perhatikan pada `.iloc[4]`, angka 4 diambil berdasarkan nomor index data. dari kode dibawah ini kita berhasil menampilkan umur dari data index 4 (data ke 5)

```python
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print("Age is :", csv['Age'].iloc[4])
print("----- DATASET -----")
print(csv.head())
```

✓ 0.8s Python

```
Age is : 31
----- DATASET -----
   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

selain itu kita dapat menampilkan data dari range tertentu, misal kita menampilkan data dari index 5 ke index 10. kita bisa menggunakan `print(csv.iloc[5:10])`

Menampilkan Informasi statistik.

pada praktik ini kita akan memanfaatkan fungsi .describe() pada pandas. fungsi ini digunakan untuk melihat beberapa detail statistik dasar seperti persentil, mean, std, dll. dari dataFrame atau serangkaian nilai numeric.

```
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
print(csv.describe(exclude=['O']))
```

✓ 1.1s Python

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Melakukan Pengecekan Data NULL / NAN pada data
dengan memanfaatkan fungsi yang disediakan oleh pandas kita dapat melakukan pengecekan data jika data itu memiliki nilai null atau nan. tentusaja akan sangat mempermudah ketika kita memiliki ribuan atau lebih data, tentusaja jika dilakukan manual akan sangat memakan waktu dan tenaga.

```
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data_missingvalue.csv")
print(csv.isnull().values.any())
```

✓ 1.7s Python

```
True
```

jika dilihat dari dokumentasinya, fungsi ini akan mengembalikan boolean yang menunjukkan jika nilainya NA. Nilai NA, seperti None atau numpy.NaN (NaN / Not a Number), dipetakan ke nilai True. Segala sesuatu yang lain dipetakan ke nilai False. Karakter seperti string kosong ' ' atau numpy.inf tidak dianggap sebagai nilai NA (kecuali jika Anda menyetel pandas.options.mode.use_inf_as_na = True).

lalu untuk mengatasi kekosongan data diatas kita dapat mengisina dengan MEAN atau MEDIAN, seperti pada contoh yang ada. kode praktik ada pada halama selanjutnya.

Mean

```python
import pandas as pd

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data_missingvalue.csv")
print(csv.mean(numeric_only=True))
print("--- DATASET TERDAPAT NILAI KOSONG ---")
print(csv.head(5))

csv=csv.fillna(csv.mean(numeric_only=True))
print("--- DATASET YANG SUDAH DISI DENGAN MEAN ---")
print(csv.head(5))
```

✓ 1.7s Python

```
CustomerID              100.500000
Age                      38.939698
Annual Income (k$)       61.005051
Spending Score (1-100)   50.489899
dtype: float64
--- DATASET TERDAPAT NILAI KOSONG ---
   CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male  19.0                15.0                    39.0
1           2    Male   NaN                15.0                    81.0
2           3  Female  20.0                 NaN                     6.0
3           4  Female  23.0                16.0                    77.0
4           5  Female  31.0                17.0                     NaN
--- DATASET YANG SUDAH DISI DENGAN MEAN ---
   CustomerID   Genre        Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male  19.000000           15.000000               39.000000
1           2    Male  38.939698           15.000000               81.000000
2           3  Female  20.000000           61.005051                6.000000
3           4  Female  23.000000           16.000000               77.000000
4           5  Female  31.000000           17.000000               50.489899
```

Median

```python
import pandas as pd

csv_data = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data_missingvalue.csv")
print("--- DATASET TERDAPAT NILAI KOSONG ---")
print(csv_data.head(5))

csv_data=csv_data.fillna(csv_data.median(numeric_only=True))
print("--- DATASET YANG SUDAH DISI DENGAN MEAN ---")
print(csv_data.head(5))
```

✓ 0.8s Python

```
--- DATASET TERDAPAT NILAI KOSONG ---
   CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male  19.0                15.0                    39.0
1           2    Male   NaN                15.0                    81.0
2           3  Female  20.0                 NaN                     6.0
3           4  Female  23.0                16.0                    77.0
4           5  Female  31.0                17.0                     NaN
--- DATASET YANG SUDAH DISI DENGAN MEAN ---
   CustomerID   Genre   Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male  19.0                15.0                    39.0
1           2    Male  36.0                15.0                    81.0
2           3  Female  20.0                62.0                     6.0
3           4  Female  23.0                16.0                    77.0
4           5  Female  31.0                17.0                    50.0
```

Praktik Normalisasi Menggunakan Scikit Learn Pada Python

Scikit Learn merupakan library pada python yang digunakan untuk machine learning dan data science. Salah satu library yang selalu menjadi favorit dan komunitasnya sangat kuat. Scikit-learn sendiri tidak hanya untuk analytics saja, namun juga untuk pre-processing, feature selection, dan proses analysis lainnya

```python
import pandas as pd
import numpy as np
from sklearn import preprocessing

csv = pd.read_csv("https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/shopping_data.csv")
array = csv.values

# Memisahkan fitur dari dataset.
X = array[:,2:5]
# Memisahkan class dari dataset
Y = array[:,0:1]

dataset = pd.DataFrame({'Customer ID':array[:,0],'Gender':array[:,1],'Age':array[:,2],'Income':array[:,3],'Spending Score':array[:,4]})

print("-- DATASET SEBELUM NORMALISASI --")
print(dataset.head(5))

# inisialisasi normalisasi MinMax
min_max_scaler = preprocessing.MinMaxScaler(feature_range=(0,1))

#transformasi MinMax untuk fitur
data = min_max_scaler.fit_transform(X)
dataset = pd.DataFrame({'Age':data[:,0],'Income':data[:,1],'Spending Score':data[:,2],'Customer ID':array[:,0],'Gender':array[:,1]})

print("-- DATASET SETELAH NORMALISASI --")
print(dataset.head(5))
```

✓ 2.5s Python

Hasil Output

```
-- DATASET SEBELUM NORMALISASI --
   Customer ID  Gender Age Income Spending Score
0           1    Male  19     15             39
1           2    Male  21     15             81
2           3  Female  20     16              6
3           4  Female  23     16             77
4           5  Female  31     17             40
-- DATASET SETELAH NORMALISASI --
        Age    Income  Spending Score Customer ID  Gender
0  0.019231  0.000000        0.387755           1    Male
1  0.057692  0.000000        0.816327           2    Male
2  0.038462  0.008197        0.051020           3  Female
3  0.096154  0.008197        0.775510           4  Female
4  0.250000  0.016393        0.397959           5  Female
```