MODUL PRAKTIKUM 2 - PEMBELAJARAN MESIN LOAD DATA

Metode yang bisa digunakan untuk membaca sebuah data pada phyton dengan beberapa format yang umum digunakan seperti .csv, .txt, .xls/.xlsx dan juga image. Saat akan mengimpor file eksternal, ada beberapa poin yang harus diperhatikan diantaranya :

- Periksa apakah baris tajuk ada atau tidak
- Perlakuan nilai khusus sebagai nilai yang hilang
- Jenis data yang konsisten dalam variabel (kolom)
- Variabel Jenis Tanggal dalam format tanggal yang konsisten.
- Tidak ada pemotongan baris saat membaca data eksternal

1. Pendahuluan

Apa itu Pandas ?

- Pandas adalah library Python yang digunakan untuk bekerja dengan kumpulan data.
- Pandas memiliki fungsi untuk menganalisis, membersihkan, menjelajahi, dan memanipulasi data.
- Nama "Pandas" memiliki referensi ke "Panel Data", dan "Python Data Analysis" dan dibuat oleh Wes McKinney pada tahun 2008.

Fitur Kunci dari Pandas ?

- Objek DataFrame yang cepat dan efisien dengan pengindeksan default dan disesuaikan
- Alat untuk memuat data ke dalam objek data di memori dari format file yang berbeda.
- Penyelarasan data dan penanganan terintegrasi untuk data yang hilang.
- Pembentukan ulang dan perputaran set tanggal.
- Pemotongan, pengindeksan, dan subset berbasis label dari kumpulan data besar.
- Kolom dari struktur data dapat dihapus atau disisipkan.
- Kelompokkan menurut data untuk agregasi dan transformasi.
- Kelompokkan menurut data untuk agregasi dan transformasi.
- Kelompokkan menurut data untuk agregasi dan transformasi.

2. Memulai Pandas

Instalasi Pandas

Jika Anda menggunakan **Anaconda**, pandas harusnya sudah terpasang secara otomatis. String versi disimpan di bawah atribut **version**.

Import Pandas

Setelah Pandas dipasang, impor ke aplikasi Anda dengan menambahkan kata kunci **import**:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive",
```

```
import pandas
```

Sekarang Panda diimpor dan siap digunakan.

Mengecek versi Pandas

```
print(pandas.__version__)
```

```
1.3.5
```

Pandas as pd

Panda biasanya diimpor dengan alias pd. alias: Dalam Python alias adalah nama alternative untuk merujuk pada hal yang sama. Buat alias dengan kata kunci **as** saat mengimpor:

```
import pandas as pd
```

Sekarang paket Pandas bisa disebut sebagai **pd**, bukan **pandas**.

3. Pandas DataFrames

Apa itu DataFrame?

Pandas DataFrame adalah struktur data 2 dimensi, seperti larik 2 dimensi, atau tabel dengan baris dan kolom. Fitur dari dataframe:

- Kolom berpotensi memiliki tipe yang berbeda
- Ukuran - Dapat Diubah
- Sumbu berlabel (baris dan kolom)
- Dapat Melakukan operasi Aritmatika pada baris dan kolom

Pandas.DataFrame

DataFrame pandas dapat dibuat menggunakan konstruktor berikut – **pandas.DataFrame(data, index, columns, dtype, copy)**

Keterangan:

- data - data mengambil berbagai bentuk seperti ndarray, series, map, list, dict, constants dan juga DataFrame lainnya.
- index - Untuk label baris, Indeks yang akan digunakan untuk frame yang dihasilkan
- adalah Opsional Default np.arange (n) jika tidak ada indeks yang dilewatkan
- kolom - Untuk label kolom, sintaks default opsional adalah - np.arange (n). Ini hanya benar jika tidak ada indeks yang dilewatkan.
- dtype - Tipe data setiap kolom.
- copy - Perintah ini (atau apa pun itu) digunakan untuk menyalin data, jika defaultnya adalah False.

Membuat DataFrame

DataFrame pandas dapat dibuat menggunakan berbagai input seperti -

- list
- dict
- Series
- Numpy ndarrays
- DataFrame lain

Membuat DataFrame dari List

```
import pandas as pd
data = [['Shankara',6],['Alvarendra',3],['Rakasulung', 20]]
df = pd.DataFrame(data,columns=['Name','Age'])
print(df)
```

```
          Name  Age
0    Shankara    6
1  Alvarendra    3
2  Rakasulung   20
```

Membuat DataFrame dari Dictionary

```
import pandas as pd

data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
#Load data into a DataFrame Object:
df = pd.DataFrame(data)

print(df)
```

```
   calories  duration
0       420        50
1       380        40
2       390        45
```

4. Pandas Read .csv

Cara sederhana untuk menyimpan kumpulan data besar adalah dengan menggunakan file .csv (file yang dipisahkan koma). File .csv berisi teks biasa dan merupakan format terkenal yang dapat dibaca oleh semua orang termasuk Panda. Sebagai contoh, akan menggunakan file .csv yang disebut data.csv.

Membaca file .csv dengan header row

Ini adalah sintaks dasar dari fungsi read_csv (). Anda hanya perlu menyebutkan nama file. Ini mengasumsikan Anda memiliki nama kolom di baris pertama file .csv Anda.

```
data = pd.read_csv("/content/drive/MyDrive/MATERI/Pembelajaran Mesin/Praktikum Genap 20212022/data.csv")
data
```

Ini menyimpan data dengan cara yang seharusnya seperti kita memiliki header di baris pertama file data kita. Penting untuk disoroti bahwa header = 0 adalah nilai default. Karenanya kita tidak perlu menyebutkan parameter header = 0. Artinya header dimulai dari baris pertama karena pengindeksan python dimulai dari 0. Kode di atas setara dengan baris kode ini. **pd.read_csv ("data.csv", header = 0).**

Menentukan nama kolom Anda sendiri, bukan baris header dari file .csv

```
dataNew = pd.read_csv("/content/drive/MyDrive/MATERI/Pembelajaran Mesin/Praktikum Genap 20212022/data.csv",
                      skiprows=1, names=['Durasi', 'Nadi', 'MaxNadi', 'Kalori'])
dataNew
```

169 rows × 4 columns

skiprows = 1 berarti kita mengabaikan baris pertama dan names = opsi digunakan untuk menetapkan nama variabel secara manual.

5. Memahami Data

Melihat Data

```
peek = data.head(20)
print(peek)
```

```
    Duration  Pulse  Maxpulse  Calories
0         60    110       130     409.1
1         60    117       145     479.0
2         60    103       135     340.0
3         45    109       175     282.4
4         45    117       148     406.0
5         60    102       127     300.0
6         60    110       136     374.0
7         45    104       134     253.3
8         30    109       133     195.1
9         60     98       124     269.0
10        60    103       147     329.3
11        60    100       120     250.7
12        60    106       128     345.3
13        60    104       132     379.3
14        60     98       123     275.0
15        60     98       120     215.2
16        60    100       120     300.0
17        45     90       112       NaN
```

```
18        60    103       123     323.0
19        45     97       125     243.0
```

Dimensi Data

```
shape = data.shape
print(shape)
```

```
(169, 4)
```

Tipe Data antar Atribut

```
types = data.dtypes
print(types)
```

```
Duration      int64
Pulse         int64
Maxpulse      int64
Calories    float64
dtype: object
```

Statistik Deskriptif

```
description = data.describe()
print(description)
```

```
             Duration       Pulse    Maxpulse     Calories
count  169.000000  169.000000  169.000000   164.000000
mean    63.846154  107.461538  134.047337   375.790244
std     42.299949   14.510259   16.450434   266.379919
min     15.000000   80.000000  100.000000    50.300000
25%     45.000000  100.000000  124.000000   250.925000
50%     60.000000  105.000000  131.000000   318.600000
75%     60.000000  111.000000  141.000000   387.600000
max    300.000000  159.000000  184.000000  1860.400000
```

Korelasi antar Atribut

```
correlations = data.corr(method='pearson')
print(correlations)
```

```
             Duration     Pulse  Maxpulse  Calories
Duration  1.000000 -0.155408  0.009403  0.922717
Pulse    -0.155408  1.000000  0.786535  0.025121
Maxpulse  0.009403  0.786535  1.000000  0.203813
Calories  0.922717  0.025121  0.203813  1.000000
```

✓ 0s completed at 9:06 AM