

BAB I

RUMUSAN MASALAH DAN TUJUAN

1.1 Rumusan Masalah

1. Bagaimana cara meng-import library pada python?
2. Data apa yang digunakan untuk dimuat dan dilakukan summary?
3. Bagaimana cara memuat dataset yang telah ditentukan?
4. Bagaimana menentukan dimensi dari dataset?
5. Bagaimana cara melihat isi dataset?
6. Bagaimana cara melihat distribusi kelas data?
7. Bagaimana cara melihat ringkasan statistik data?
8. Bagaimana cara visualisasi data menggunakan plot univariat?
9. Bagaimana cara visualisasi data menggunakan plot multivariat?

1.2 Tujuan

1. Mengetahui cara meng-import library pada python.
2. Mengetahui data apa yang digunakan untuk dimuat dan dilakukan summary.
3. Mengetahui cara memuat dataset yang telah ditentukan.
4. Mengetahui cara menentukan dimensi dari dataset.
5. Mengetahui cara melihat isi dataset.
6. Mengetahui cara melihat distribusi kelas data.
7. Mengetahui cara melihat ringkasan statistik data.
8. Mengetahui cara visualisasi data menggunakan plot univariat.
9. Mengetahui cara visualisasi data menggunakan plot multivariat.

BAB II

DASAR TEORI

2.1 Pandas Python Library

Pandas adalah sebuah librari berlisensi BSD dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan dan berkinerja tinggi untuk bahasa pemrograman Python.

Dengan kata lain, Pandas adalah librari analisis data yang memiliki struktur data yang diperlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang cocok untuk analisis (yaitu tabel). Pandas melakukan tugas penting seperti menyelaraskan data untuk perbandingan dan penggabungan set data, penanganan data yang hilang, dll, itu telah menjadi sebuah librari de facto untuk pemrosesan data tingkat tinggi dalam Python (yaitu statistik).

2.2 Matplotlib

Matplotlib adalah library Python 2D yang dapat menghasilkan plot dengan kualitas tinggi dalam berbagai format dan dapat digunakan di banyak platform.

Matplotlib dapat digunakan sebagai pembuat grafik dalam berbagai platform, seperti Python dan Jupyter. Grafik yang dapat dibuat beragam, seperti grafik garis, batang, lingkaran, histogram, dsb.

2.2 Scatter Diagram

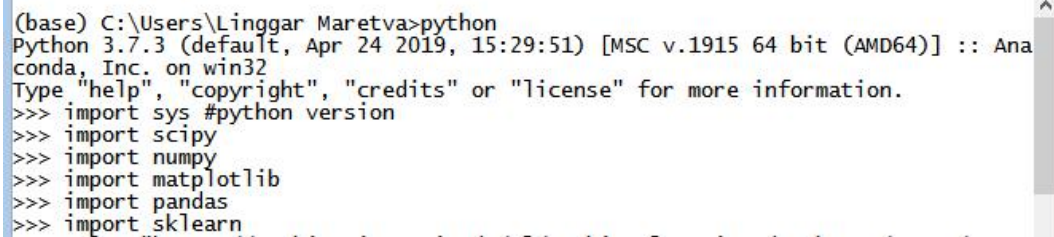
Scatter matriks adalah statistik yang berfungsi untuk melakukan pengujian terhadap seberapa kuatnya hubungan antara 2 (dua) variabel serta menentukan jenis hubungan dari 2 (dua) variabel tersebut apakah hubungan Positif, hubungan Negatif ataupun tidak ada hubungan sama sekali. Bentuk dari Scatter Diagram adalah gambaran grafis yang terdiri dari sekumpulan titik-titik (point) dari nilai sepasang variabel (Variabel X dan Variabel Y).

BAB III

PEMBAHASAN

3.1 Import Library Pada Python

Pertama, kita import terlebih dahulu library - library yang akan kita gunakan pada program python kita, untuk mengolah dataset yang kita miliki.



```
(base) C:\Users\Linggar Maretva>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Ana
conda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys #python version
>>> import scipy
>>> import numpy
>>> import matplotlib
>>> import pandas
>>> import sklearn
```

Seperti yang ditunjukkan pada gambar di atas, library yang kita import adalah Scipy, Numpy, Matplotlib, Pandas, dan Sklearn.

3.2 Menentukan Data Yang Digunakan

Pada laporan praktikum ini, saya menggunakan dataset yang didapat dari Repository UCI Machine Learning, yaitu Haberman's Survival Data Set. Dataset ini berisi kasus-kasus dari penelitian yang dilakukan antara tahun 1958 dan 1970 di Rumah Sakit Billings University of Chicago tentang kelangsungan hidup pasien yang telah menjalani operasi untuk kanker payudara.

Dataset ini memiliki 4 attribut. Yang pertama adalah attribut age, yaitu usia pasien saat melakukan operasi, kemudian attribut year, yaitu tahun ketika operasi dilakukan, kemudian yang ketiga attribut positive-axilliary, yaitu jumlah noda positive-axilliary yang terdeteksi. Kemudian yang keempat adalah attribut kelas, yaitu attribut survival, yaitu status waktu bertahan hidup pasien, nilai 1 untuk yang pasien bertahan 5 tahun atau lebih dan nilai 2 untuk pasien yang meninggal dalam 5 tahun. Semua attribut ini berjenis numerical.

3.3 Memuat Dataset

Dataset dapat dimuat dengan mengambil langsung dari alamat repository UCI Machine Learning dengan cara mengeksekusi script sebagai berikut.

```
>>> url = "https://archive.ics.uci.edu/ml/machine-learning-databases/haberman/haberman.data"
>>> names = ['age', 'year', 'positive-axillary', 'class']
>>> dataset = pandas.read_csv(url, names=names)
```

Data diambil dengan dimasukkan ke variabel url, kemudian kita definisikan nama - nama kolom pada dataset yang akan kita gunakan pada variabel names. Lalu kita baca dengan library Pandas dengan inputan url dan names.

3.4 Menentukan Dimensi Dari Dataset

Dimensi dari dataset merupakan gambaran singkat mengenai banyaknya jumlah baris yang menunjukkan banyaknya sampel data dan jumlah kolom yang menunjukkan atribut data dari dataset. Dari dataset yang digunakan, dimensi yang dihasilkan adalah sebagai berikut.

```
>>> print(dataset.shape)
(306, 4)
```

Dari data di atas, dimensi yang didapat adalah 306 sampel data, dan 4 atribut data.

3.5 Melihat Isi Dataset

Kita akan mencoba untuk menampilkan 20 baris pertama dari dataset. Dapat kita lakukan dengan script dan hasil sebagai berikut

```
>>> print(dataset.head(20))
   age  year  positive-axillary  class
0    30    64                1      1
1    30    62                3      1
2    30    65                0      1
3    31    59                2      1
4    31    65                4      1
5    33    58               10      1
6    33    60                0      1
7    34    59                0      2
8    34    66                9      2
9    34    58               30      1
10   34    60                1      1
11   34    61               10      1
12   34    67                7      1
13   34    60                0      1
14   35    64               13      1
15   35    63                0      1
16   36    60                1      1
17   36    69                0      1
18   37    60                0      1
19   37    63                0      1
```

3.6 Distribusi Kelas Data

Distribusi dari kelas data merupakan jumlah dari data dari tiap kelas yang ada. Karena kelas yang ada hanya ada dua, yaitu kelas 1 dan 2, maka hasilnya dapat kita eksekusi dan kita lihat hasilnya sebagai berikut.

```
>>> print(dataset.groupby('class').size())
class
1      225
2       81
dtype: int64
```

Dari data yang ditampilkan, kita dapatkan bahwa dari kelas 1, atau kelas dimana pasien dapat survive 5 tahun atau lebih ada 225 data. Sedangkan untuk kelas 2, atau kelas dimana pasien meninggal dibawah 5 tahun ada 81 data.

3.7 Ringkasan Statistik Data

Kita akan mencoba untuk menampilkan ringkasan statistik dari dataset yang kita gunakan. Data yang coba kita tampilkan adalah count, mean, standard deviasi, nilai min, nilai max, dan quartile. Untuk menampilkan ringkasan statistik tersebut, dapat kita lakukan dengan eksekusi dan hasil eksekusi sebagai berikut.

```
>>> print(dataset.describe())
```

	age	year	positive-axillary	class
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

Ringkasan statistik yang didapat ditampilkan dengan kolom berupa atribut data yang ada, yaitu age, year, positive-axillary, dan class.

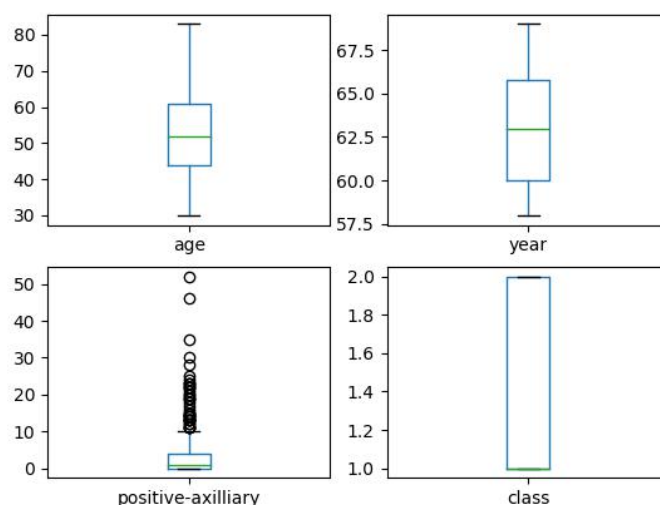
3.8 Visualisasi Data Menggunakan Plot Univariat

Plot univariat adalah plot dari masing-masing variabel individu. Mengingat bahwa variabel inputnya numerik, kita bisa membuat jenis plot box.

Untuk membuat plot, dapat kita eksekusi dengan script sebagai berikut

```
>>> import matplotlib.pyplot as plt
>>> dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
age
year
positive-axillary
class
dtype: object
>>> plt.show()
```

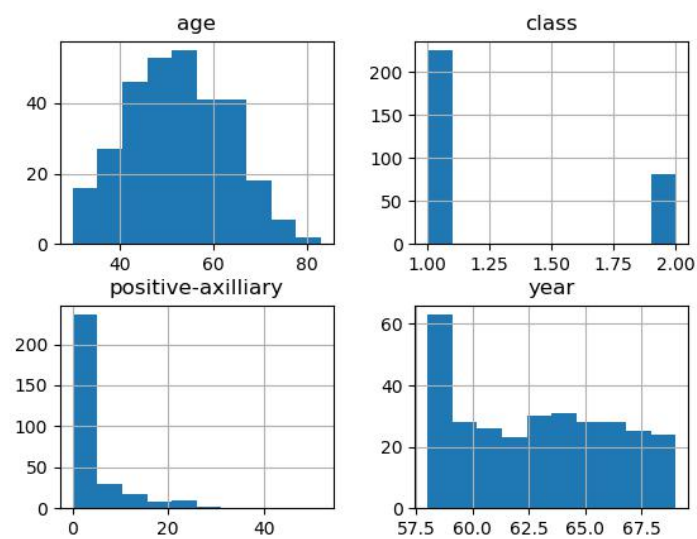
Script pada baris terakhir yaitu plt.show() akan menampilkan hasil plot yang dibuat, yang berjenis box. Outputnya sebagai berikut.



Selain itu, kita juga dapat membuat histogram untuk mendapatkan ide distribusi dari atribut - atribut yang ada. Untuk melakukannya, dapat kita lakukan dengan mengeksekusi script berikut.

```
>>> dataset.hist()
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24EF229E8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F0C7B00>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F6B30F0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F6E66A0>]],
      dtype=object)
>>> plt.show()
```

Dari script di atas, akan menghasilkan output histogram sebagai berikut.



3.9 Visualisasi Data Menggunakan Plot Multivariat

Selanjutnya kita bisa melihat interaksi antar variabel. Pertama, kita lihat scatterplots dari semua pasang atribut. Hal ini dapat membantu melihat hubungan terstruktur antara variabel input.

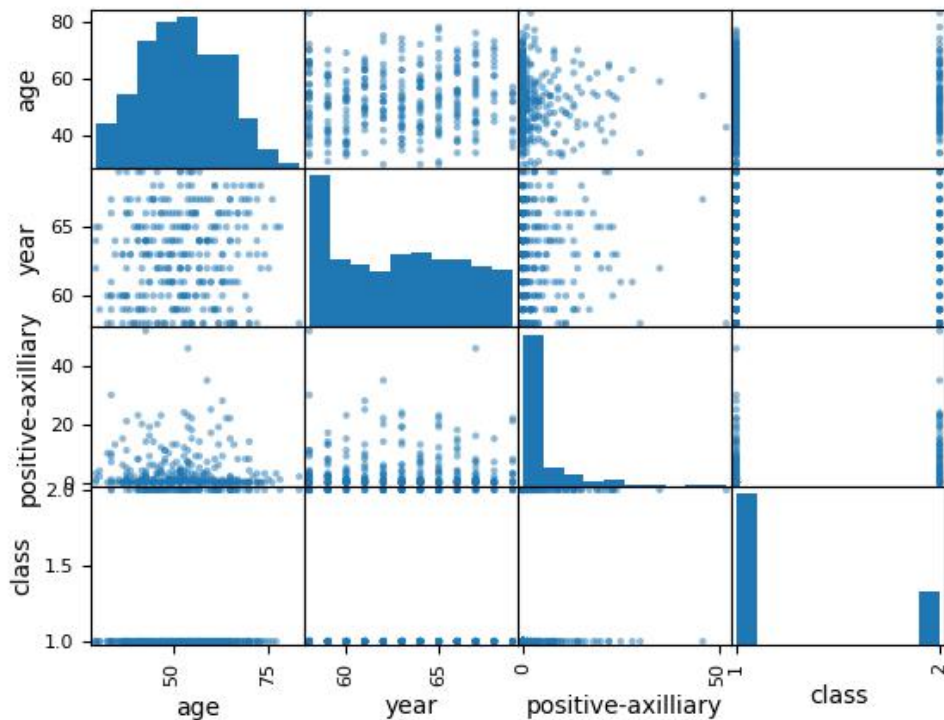
Untuk melihat hasil scatterplots-nya, maka dapat kita lakukan dengan mengeksekusi script berikut ini.


```

>>> import pandas
>>> from pandas.plotting import scatter_matrix
>>> scatter_matrix(dataset)
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F40A240>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F468B70>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F4A3160>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F4D5710>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F507CC0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F5452B0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F575860>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F5A7E48>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F5A7E80>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F6169B0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F648F60>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F805550>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F835B00>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F8760F0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F8E76A0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000000A24F918C50>]],
      dtype=object)
>>> plt.show()

```

Output dari script di atas adalah sebagai berikut.



BAB IV

KESIMPULAN

A. Kesimpulan

Untuk mengolah data yang kita miliki di Python, kita memerlukan beberapa library. Beberapa library yang digunakan adalah Pandas dan Matplotlib.

Kemudian dengan library tersebut, kita dapat mengolah dataset yang didapat dari Repositori UCI Machine Learning, dalam laporan ini, menggunakan Haberman's Survival Data Set.

Pengolahan data yang dilakukan, kita dapat membuat grafik seperti Plot Univariat, yaitu plot dari masing-masing variabel individu, dan juga Plot Multivariat, yaitu plot dengan menggunakan scatterplot atau scatter-matrix.

DAFTAR PUSTAKA

Informatics (14 Oktober 2019). "*Pertemuan 1 - Python Introduction*". Makalah disajikan dalam Praktikum Pembelajaran Mesin di laboratorium komputer gedung E Universitas Diponegoro. Semarang, 14 Oktober 2019.

IIN MUTMAINNAH (6 Januari 2019). "*Mengenal Pandas Dalam Python*". Retrieved 18 Oktober 2019. from Medium : <https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40>

Syauqi Muhammad Dhiya Ulhaq (12 Desember 2018). "*Visualisasi Data Menggunakan Python dan Matplotlib*". Retrieved 18 Oktober 2019. from Medium : <https://medium.com/@symdu31/visualisasi-data-menggunakan-python-dan-matplotlib-de271812b2fd>