



BADAN PENGKAJIAN DAN PENERAPAN TEKNOLOGI

PROGRAM
INOVASI TEKNOLOGI INTELLIGENT COMPUTING
BERBASIS BIOMETRIK DAN TEKNOLOGI BAHASA

TECHNICAL NOTE

No. 01/TN/ES 1.1.3/IIC/I/2017

Penyajian Data dengan Python

WP 1.1

Pengembangan Algoritma Identifikasi Memakai Multimodal Biometrik

WBS 1

Teknologi Biometrik

Dibuat oleh :	Diperiksa oleh :	Disetujui oleh :
Engineering Staff	Leader	Group Leader
Nama : Rachmawan Atmaji P., S.Kom.	Nama : Rully Kusumajaya, S. Kom.	Nama : Dr. Anto Satriyo Nugroho
Tgl : 27 Januari 2017	Tgl : 30 Januari 2017	Tgl : 31 Januari 2017

Selama ini, penyajian data untuk kegiatan Intelligent Computing, terutama terkait dengan pengembangan dan pengujian system biometric (modal fingerprint) dilakukan dengan menggunakan software Microsoft Excel. Kali ini akan dibuat dengan Bahasa pemrograman Python agar penyajiannya lebih baik, jelas dan rapi. Salah satu kelebihan dari penyajian data dengan Python ini adalah, dapat melakukan zooming data dengan mudah dengan resolusi yang baik.

Adapun grafik-grafik yang akan dibuat dalam Bahasa pemrograman Python ini adalah kurva ROC dan Boxplot.

1. Kurva ROC

Kurva Receiver Operating Characteristic (ROC) adalah plot grafis yang mengilustrasikan performa dari system klasifier binary saat threshold diskriminannya bervariasi. Input dari kurva ROC adalah nilai FAR dan FRR yang berpasangan. Jumlah baris dari FAR dan FRR harus sama. Jika tidak, maka akan terjadi error. Perhitungan FAR dan FRR dari pengujian system biometric telah dilakukan oleh aplikasi lain.

Yang pertama dilakukan adalah membaca data dari excel, kemudian menyimpannya dalam struktur data array FAR dan FRR. Kemudian mem-plot array FRR dalam sumbu-x, dan array FAR dalam sumbu-y. Plotting dilakukan dengan library Matplotlib. Berikut adalah source code dan contoh hasil output running file.

```
import xlrd
import matplotlib.pyplot as plt

csfont = {'fontname':'Comic Sans MS'}
hfont = {'fontname':'Tahoma'}
afont = {'fontname':'Arial'}

workbook = xlrd.open_workbook('D:/Tugas PTK/Biometrik/Pengujian
Sistem Biometrik/Vendor 16/Rekap.xlsx')
worksheet = workbook.sheet_by_name('FARvsFRR_03')
num_rows = worksheet.nrows - 1
curr_row = 0

#creates an array to store all the rows

far_array = []
frr_array = []

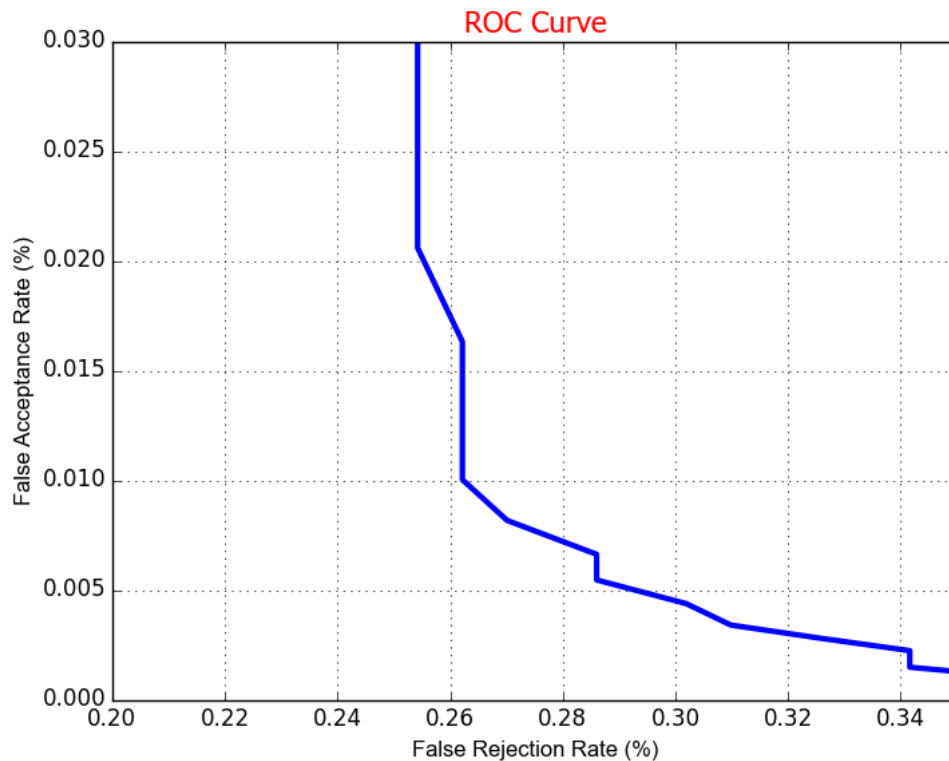
while curr_row < num_rows:
    value_far = worksheet.cell(curr_row,0).value
    value_frr = worksheet.cell(curr_row,1).value
    far_array.append(value_far*100)
    frr_array.append(value_frr*100)
    curr_row += 1

print far_array
print frr_array

[float(i) for i in far_array]
[float(i) for i in frr_array]

plt.plot(frr_array, far_array, '-b', linewidth=3.0)
```

```
#plt.axis([0, 100, 0, 100])
plt.axis([0, 5, 0, 0.03])
plt.grid(True)
plt.title("ROC Curve", fontsize=16, color='red', **hfont)
plt.xlabel('False Rejection Rate (%)', fontsize=12, color='black',
**afont)
plt.ylabel('False Acceptance Rate (%)', fontsize=12, color='black',
**afont)
plt.show()
```



2. Boxplot

Box plot adalah metode yang baik untuk menampilkan kelompok dari data-data numerik melalui quartil-quartil mereka. Box plot dapat memiliki garis yang bertambah secara vertical dari kotak-kotak (whisker) mengindikasikan keberagaman di luar quartil atas dan bawah, sehingga menghasilkan istilah box-and-whisker-plot dan box-and-whisker diagram. Outlier dapat diplotting sebagai titik-titik individual. Bagian bawah kotak selalu quartil pertama, bagian atas kotak adalah quartil ketiga, dan pita di dalam kotak adalah quartil kedua (median).

Boxplot dalam scenario pengujian kinerja system Biometrik, digunakan untuk menggambarkan deviasi/selisih hasil pemadanan terhadap input yang sama pada dua buah system biometric. Pada pengujian kinerja Boxplot ini digunakan untuk mencari selisih antara hasil pemadanan pada device dengan pemadanan pada PC. Berikut adalah source code dan contoh hasil output running file.

```

import csv
import math
from collections import defaultdict
import matplotlib.pyplot as plt
import numpy as np

columns = defaultdict(list) # each value in each column is appended
to a list

afont = {'fontname': 'Arial'}

f = open('D:/Tugas PTK/Biometrik/Pengujian Sistem Biometrik/Vendor
16/frr_sken2_vendor16_final_new.csv', 'rb') # opens the csv file
try:
    reader = csv.DictReader(f, delimiter=',') # read rows into a
dictionary format
    for row in reader: # read a row as {column1: value1, column2:
value2,...}
        for (k,v) in row.items(): # go over each column name and
value
            columns[k].append(v) # append the value into the
appropriate list
                                # based on column name k

    #print(columns['DUT Score'])
    #print(columns['Live_Score'])

    arr_live_score = []
    arr_dut_score = []

    arr_deviation_x = []
    arr_deviation_y = []

    arr_norm = []

    datacount = len(columns['Live_Score'])

    for x in range(0, datacount):
        arr_live_score.append(int(columns['Live_Score'][x]))

    for y in range(0, datacount):
        arr_dut_score.append(int(columns['DUT_Score'][y]))

    for z in range(0, datacount):

arr_deviation_x.append(float((arr_live_score[z]+arr_dut_score[z])/mat
h.sqrt(2)))

        for z in range(0, datacount):
            arr_deviation_y.append(float((arr_dut_score[z]-
arr_live_score[z])/math.sqrt(2)))

        for z in range(0, datacount):
            arr_norm.append(abs(arr_deviation_x[z]))

    fig = plt.figure()
    ax = fig.add_subplot(111)

    #print arr_norm

    p1 = np.percentile(arr_norm, 25);
    p2 = np.percentile(arr_norm, 50);

```

```

p3 = np.percentile(arr_norm, 75);

str_p1 = 'Q1 : %.3f' % p1
str_p2 = 'Q2 : %.3f' % p2
str_p3 = 'Q3 : %.3f' % p3

plt.text(0.3, 0.138, str_p1, ha='center', va='center',
transform=ax.transAxes)
plt.text(0.3, 0.225, str_p2, ha='center', va='center',
transform=ax.transAxes)
plt.text(0.3, 0.385, str_p3, ha='center', va='center',
transform=ax.transAxes)
plt.text(0.45, 0.8, 'Outlier', ha='center', va='center',
color='red', transform=ax.transAxes)

B = ax.boxplot(arr_norm, sym='+', vert=True, whis=1.0,
showfliers=True)

C = [item.get_ydata() for item in B['whiskers']]

str_pdb = 'PDB : %.3f' % C[0][1]
str_pda = 'PDA : %.3f' % C[1][1]

plt.text(0.36, 0.04, str_pdb, ha='center', va='center',
transform=ax.transAxes)
plt.text(0.36, 0.63, str_pda, ha='center', va='center',
transform=ax.transAxes)
plt.ylabel('Deviasi (Jarak titik ke garis y=x)', fontsize=12,
color='black', **afont)

plt.show()
finally:
f.close()          # closing

```

