

Laporan Tugas Pemrograman Learning



Dosen Pembimbing:
IZZATUL UMMAH,S.T., M.T.
CII2M3-IF-44-11

Disusun oleh:
1301200240 Ramadhan Aditya Ibrahim
1301204125 Ryan Chandra Hadi

S1 Informatika
Universitas Telkom
Bandung, Jawa Barat 2022

Kata Pengantar

Dengan mengucapkan puji dan rasa syukur kepada Allah SWT yang telah memberikan rahmat sehingga kita dapat menyelesaikan tugas dari mata kuliah Pengantar Kecerdasan Buatan dengan tema “Learning” dengan benar dan tepat waktu.

Untuk memenuhi nilai tugas pada mata kuliah Pengantar Kecerdasan Buatan, maka dibuatkan tugas yang dapat kita selesaikan. Tidak hanya itu, tujuan dari pembuatan laporan dan pengerjaan tugas ini adalah untuk menambah wawasan tentang pembahasan Learning bagi kita semua.

Kami mengucapkan terima kasih kepada semua pihak dari mulai ibu Izzatul Ummah selaku dosen pembimbing yang telah memberikan kita tugas besar untuk membuat Algoritma tentang learning menggunakan metode K-NN.

Kami sangat menyadari laporan yang kami susun masih jauh dari kata sempurna. Tetapi kita akan terus berusaha untuk selalu menjadi lebih baik untuk kedepannya.

Bandung, 16 Juni 2022

Kelompok 6

BAB I

PENDAHULUAN

1. PERSOALAN

1.1 DESKRIPSI TUGAS

Diberikan file `traintest.xlsx` yang terdiri dari dua sheet: `train` dan `test`, yang berisi dataset untuk problem klasifikasi biner (binary classification). Setiap record atau baris data dalam dataset tersebut secara umum terdiri dari nomor baris data (*id*), fitur input (x_1 sampai x_3), dan output kelas (y). Fitur input terdiri dari nilai-nilai integer dalam range tertentu untuk setiap fitur. Sedangkan output kelas bernilai biner (0 atau 1).

id	x1	x2	x3	y
1	60	64	0	1
2	54	60	11	0
3	65	62	22	0
4	34	60	0	1
5	38	69	21	0

Sheet `train` berisi 296 baris data, lengkap dengan target output kelas (y). Gunakan sheet ini untuk tahap pemodelan atau pelatihan (training) model sesuai metode yang Anda gunakan. Adapun sheet `test` berisi 10 baris data, dengan output kelas (y) yang disembunyikan. Gunakan sheet ini untuk tahap pengujian (testing) model yang sudah dilatih. Nantinya output program Anda untuk data uji ini akan dicocokkan dengan target atau kelas sesungguhnya.

Pilihan metode yang dapat Anda gunakan (pilih salah satu)

- Decision Tree (ID3)
- KNN
- Naïve Bayes

Catatan: Anda harus mengimplementasikan metode yang Anda pilih secara lengkap untuk studi kasus/dataset yang disediakan. Urutan proses secara detail dituliskan di dalam Laporan Tugas.

Proses yang harus Anda implementasikan ke dalam program (bisa berbentuk fungsi/prosedur):

- Membaca data latih/uji
- Pelatihan atau training model
- Menyimpan model hasil training
- Pengujian atau testing model
- Evaluasi model
- Menyimpan output ke file

Catatan: Proses-proses terkait training, testing, dan evaluasi dibangun tanpa Library

1.2 OUTPUT PROGRAM

Program Anda secara umum memiliki dua tahap: pelatihan (training) dan pengujian (testing). Pada tahap training, dihasilkan output berupa model sesuai metode yang Anda gunakan. Sedangkan pada tahap testing, dihasilkan output berupa kelas (0 atau 1); Lebih jauh lagi, jika ada lebih dari satu record/baris sebagai input data untuk tahap testing, maka program dapat mengeluarkan list output kelas yang bersesuaian dengan setiap baris data testing tersebut.

BAB 2

PEMBAHASAN

2.1 Metode K-nearest neighbors (KNN)

K-nearest neighbors atau knn adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbors*). Dengan k merupakan banyaknya tetangga terdekat. Salah satu alasan kita memakai metode ini karena metode ini sangat paling dimengerti.

2.2 Membaca data latih/uji

```
df = pd.read_excel("https://raw.githubusercontent.com/bloodsking/Learning-AI/main/trainetest.xlsx", "train")
df1 = pd.read_excel("https://raw.githubusercontent.com/bloodsking/Learning-AI/main/trainetest.xlsx", "test")
```

2.3 Pelatihan atau training model

```
z = []
temp = 0
for i in range(len(x6)):
    final = 0
    final1 = 0

    jarak = euclideanV(x6, x9, x7, x10, x8, x11, i)
    jarak.sort()
    jarak = jarak[:3]
    for k in range(3):
        if(jarak[k][2]) == 1:
            final1 += 1
        else:
            final += 1

    if(final < final1):
        z.append(1)
    else:
        z.append(0)
print(z)
print(x12)
for j in range(len(x12)):
    if z[j] == x12[j]:
        temp += 1
persen = temp/len(x12) * 100
print("hasil validasi: ", persen, "%")
```

```
for zz in range(1, 40, 2):
    z = []
    temp = 0
    for i in range(len(x6)):

        final = 0
        final1 = 0

        jarak = euclideanA(x6, x9, x7, x10, x8, x11, i, len(x10))
        jarak.sort()
        jarak = jarak[:zz]
        for k in range(zz):
            if(jarak[k][2]) ==1:
                final1 += 1
            else:
                final += 1

        if(final < final1):
            z.append(1)
        else:
            z.append(0)
    print(z)
    print(x12)
    for j in range(len(x12)):
        if z[j] == x12[j]:
            temp += 1
    persen = temp/len(x12) * 100
    print("hasil validasi dari K: ", zz, "sebesar :",persen, "%\n")
```

2.7 Menyimpan output ke file

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
  y  x1 x2 x3
0  1  43 59  2
1  1  67 66  0
2  1  58 60  3
3  1  49 63  3
4  1  45 60  0
5  1  54 58  1
6  1  56 66  3
7  1  42 69  1
8  1  50 59  2
9  1  59 60  0
Data biner sudah ditambahkan ke data tes
```

BAB III

IMPLEMENTASI PROGRAM

3.1 Library

```
import pandas as pd
```

3.1.1 Pandas pada program ini berfungsi merubah data array menjadi data berbentuk tabel.

```
import math
```

3.1.2 Math pada program ini berfungsi untuk menghitung fungsi aljabar yang terdapat pada program yang kita buat.

3.2 Inisiasi

```
df = pd.read_excel("https://raw.githubusercontent.com/bloodsking/Learning-AI/main/traintest_.xlsx", "train")
df
```

	id	x1	x2	x3	y
0	1	60	64	0	1
1	2	54	60	11	0
2	3	65	62	22	0
3	4	34	60	0	1
4	5	38	69	21	0
...
291	292	59	64	1	1
292	293	65	67	0	1
293	294	53	65	12	0
294	295	57	64	1	0
295	296	54	59	7	1
296 rows × 5 columns					

```
df1= pd.read_excel("https://raw.githubusercontent.com/bloodsking/Learning-AI/main/traintest_.xlsx", "test")
df1
```

	id	x1	x2	x3	y
0	297	43	59	2	?
1	298	67	66	0	?
2	299	58	60	3	?
3	300	49	63	3	?
4	301	45	60	0	?
5	302	54	58	1	?
6	303	56	66	3	?
7	304	42	69	1	?
8	305	50	59	2	?
9	306	59	60	0	?

Pada bagian ini kita akan menampilkan data train dan data test.


```

x1 = df['x1']
x2 = df['x2']
x3 = df['x3']
x4 = df['id']
x5 = df['y']
x6 = x1.tail(80).tolist()
x7 = x2.tail(80).tolist()
x8 = x3.tail(80).tolist()
x9 = x1.head(216).tolist()
x10 = x2.head(216).tolist()
x11 = x3.head(216).tolist()
x12 = x5.tail(80).tolist()
y1 = df1['x1']
y2 = df1['x2']
y3 = df1['x3']

```

Pada bagian ini kita membagi semua data per kolom dan kami juga mengambil data tail dan head sebagai data uji coba untuk melakukan validation.

3.3 Fungsi Euclidean

Formula

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

\mathbf{p}, \mathbf{q} = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

```

def euclideanA(data1, data2, data3, data4, data5, data6, i, index):
    dataD = []
    for j in range(index):
        rumus1 = ((data1[i] - data2[j])**2 + (data3[i] - data4[j])**2 + (data5[i] - data6[j])**2)**0.5
        dataD.append(((rumus1, j, x5[j])))

    return dataD

```

Jarak Euclidean adalah formula untuk mencari jarak antara 2 titik dalam ruang dua dimensi. Pada fungsi EuclideanA itu berguna untuk mencari jarak antara 2 titik yaitu data train dan data test dan EuclideanA juga berfungsi untuk membuat data validasi yang datanya diambil dan dibagi menjadi 2 bagian.

3.4 Validation Data

```
for zz in range(1, 40, 2):
    z = []
    temp = 0
    for i in range(len(x6)):

        final = 0
        final1 = 0

        jarak = euclideanA(x6, x9, x7, x10, x8, x11, i, len(x10))
        jarak.sort()
        jarak = jarak[:zz]
        for k in range(zz):
            if(jarak[k][2]) ==1:
                final1 += 1
            else:
                final += 1

        if(final < final1):
            z.append(1)
        else:
            z.append(0)
    print(z)
    print(x12)
    for j in range(len(x12)):
        if z[j] == x12[j]:
            temp += 1
    persen = temp/len(x12) * 100
    print("hasil validasi dari K: ", zz, "sebesar :",persen, "%\n")
```

[illegible]

Pada bagian ini kita melakukan validasi data dengan membagi data train menjadi dua data train, yang terbagi menjadi 80 dan 215, data pertama kita asumsikan sebagai data test dan data kedua kita asumsikan sebagai data train. Selanjutnya kita bandingkan data hasil validasi dengan data asli dengan menggunakan persentase agar bisa melihat keberhasilan hasil validasi dengan menggunakan euclidean dan mengatur banyaknya tetangga(k) sebanyak 21 atau 23 (melihat data hasil yang terbesar).

3.5 Test Data

```
z = []
zz = 21
for i in range(len(df1)):
    final = 0
    final1 = 0
    jarak = euclideanA(y1,x1,y2,x2,y3,x3,i,len(df))
    jarak.sort()
    jarak = jarak[:zz]
    for k in range(zz):
        if(jarak[k][2]) ==1:
            final1 += 1
        else:
            final += 1

    if(final < final1):
        z.append(1)
    else:
        z.append(0)
print(z)
z1 = pd.DataFrame(z)
z1.columns = ['y']
z1['x1'] = y1
z1['x2'] = y2
z1['x3'] = y3
print(z1)

#membuat file excel
file = pd.ExcelWriter('hasil akhir.xlsx')

#memindahkan data1 ke excel
z1.to_excel(file)

#menyimpan hasil file yang sudah di export
file.save()
print("Data biner sudah ditambahkan ke data tes")
```

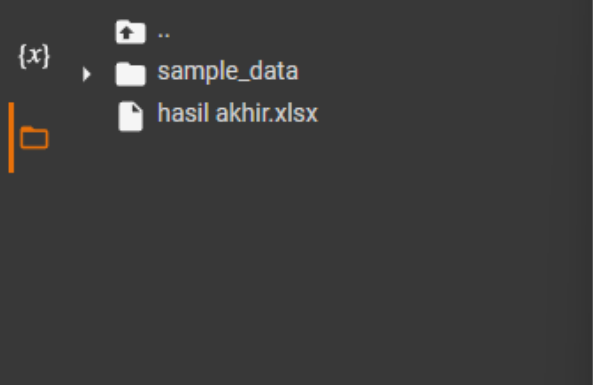
```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
  y  x1  x2  x3
0  1  43  59   2
1  1  67  66   0
2  1  58  60   3
3  1  49  63   3
4  1  45  60   0
5  1  54  58   1
6  1  56  66   3
7  1  42  69   1
8  1  50  59   2
9  1  59  60   0
Data biner sudah ditambahkan ke data tes
```

Pada bagian ini kita melakukan tes data untuk memprediksi nilai y pada data test dari data set yang diberikan dan didapatkan nilai y dari data 297 ke 306 senilai 1.

BAB IV

KESIMPULAN

Setelah kita melakukan implementasi Learning menggunakan metode K-Nearest Neighbors (K-NN) kita mendapatkan semua nilai y pada data test senilai 1.

	1		y	x1	x2	x3
	2	0	1	43	59	2
	3	1	1	67	66	0
	4	2	1	58	60	3
	5	3	1	49	63	3
	6	4	1	45	60	0
	7	5	1	54	58	1
	8	6	1	56	66	3
	9	7	1	42	69	1
	10	8	1	50	59	2
	11	9	1	59	60	0

Pembagian dalam pengerjaan tugas besar AI:

Ramadhan Aditya Ibrahim:

- Pembuatan Kode pada bagian inisiasi sampai pada fungsi euclidean.
- Pembuatan Laporan dan hasil rangkuman.

Ryan Chandra Hadi:

- Pembuatan Kode pada bagian tahap Tes, dan tahap validasi data.
- Pembuatan Laporan dan hasil rangkuman.

Link Video Youtube:

<https://youtu.be/RiUrboy4DLQ>