

Nama : Aszka Sazkia

Nim : 1227030006

PRAKTIKUM FISIKA KOMPUTASI
MODUL 11 “SUPPORT VECTOR MACHINE”

1. Kode program prediksi integral trapezoid

```
2. #Kode Program SVM
3.
4. from sklearn import svm
5.
6. #Database : Gerbang Logika AND
7. #X=Data, y=target
8. x = [[0, 0], [0, 1], [1, 0], [1,1]]
9. y = [0, 0, 0, 1]
10.
11. #training and classify
12. clf = svm.SVC()
13. clf.fit(x,y)
14.
15. #Prediksi
16. print ("Logika AND Metode Support Vector Machine (SVM)")
17. print ("Logika = Prediksi")
18. print ("0 0 =",clf.predict([[0, 0]]))
19. print ("0 1 =",clf.predict([[0, 1]]))
20. print ("1 0 =",clf.predict([[1, 0]]))
21. print ("1 1 =",clf.predict([[1, 1]]))
22.
23. # Fungsi untuk mencari integral menggunakan metode Trapezoid
24. def trapezoid(a, b, f):
25.     # Fungsi Trapezoid
26.     n = 100 # Jumlah partisi
27.     h = (b - a) / n
28.     sum = 0.0
29.     for i in range(1, n):
30.         x = a + i * h
31.         sum += f(x)
32.     integral = (h / 2) * (f(a) + 2 * sum + f(b)) # Rumus Trapezoid
33.     print(a, ",", b, ",", round(integral, 2)) # Menampilkan hasil integral
34.     return integral
35.
36. # Daftar fungsi menggunakan lambda
37. functions = [
38.     lambda x: 2 * x,          # Fungsi pertama: f(x) = 2x
39.     lambda x: 2 * x + 2,      # Fungsi kedua: f(x) = 2x + 2
40.     lambda x: 4 * x + 4,      # Fungsi ketiga: f(x) = 4x + 4
```

```

41. lambda x: 4 * x + 6, # Fungsi keempat: f(x) = 4x + 6
42. lambda x: 6 * x + 8, # Fungsi kelima: f(x) = 6x + 8
43. lambda x: 8 * x + 10, # Fungsi keenam: f(x) = 8x + 10
44. lambda x: 10 * x + 12, # Fungsi ketujuh: f(x) = 10x + 12
45. lambda x: 12 * x + 14, # Fungsi kedelapan: f(x) = 12x + 14
46. lambda x: 14 * x + 12, # Fungsi kesembilan: f(x) = 14x + 12
47. lambda x: 20 * x + 40 # Fungsi kesepuluh: f(x) = 20x + 40
48. ]
49.
50. # Looping untuk menghitung integral masing-masing fungsi
51. for i, func in enumerate(functions):
52.     a = i + 1 # Batas bawah
53.     b = i + 2 # Batas atas
54.     trapezoid(a, b, func) # Menghitung integral

```

Logika AND Metode Support Vector Machine (SVM)

Logika = Prediksi

```

0 0 = [0]
0 1 = [0]
1 0 = [0]
1 1 = [1]
1 , 2 , 3.0
2 , 3 , 7.0
3 , 4 , 18.0
4 , 5 , 24.0
5 , 6 , 41.0
6 , 7 , 62.0
7 , 8 , 87.0
8 , 9 , 116.0
9 , 10 , 145.0
10 , 11 , 250.0

```

Import library yang diperlukan

```

import numpy as np
import pandas as pd
from sklearn import svm
from google.colab import drive
import matplotlib.pyplot as plt

```

Mount Google Drive

```
drive.mount('/content/drive')
```

Path ke file Function.txt di Google Drive

```
file_path = '/content/drive/My Drive/Satu.txt' # Sesuaikan dengan path file Anda
```

Membaca data dari file

```
Database = pd.read_csv(file_path, sep=',', header=0)
```

```

# X = data, y = target
X = Database[['a', 'b']] # Pastikan kolom sesuai dengan nama yang ada di file
y = Database['Target']

# Membuat dan melatih model SVM
clf = svm.SVC()
clf.fit(X.values, y)

# Melakukan prediksi
y_pred = clf.predict(X.values)

# Menampilkan hasil prediksi
print("Hasil prediksi:")
for i, pred in enumerate(y_pred):
    print(f"({X.iloc[i, 0]}, {X.iloc[i, 1]}), {pred}")

# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)',
color='yellow')

# Menambahkan label dan judul
plt.xlabel('Indeks Data')
plt.ylabel('Nilai')
plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
plt.legend()

# Menambahkan grid
plt.grid()

plt.show()

```

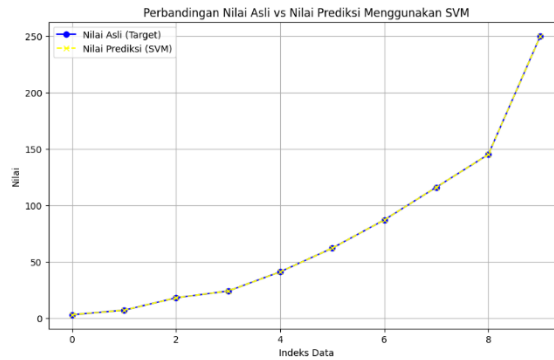
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Hasil prediksi:

```

(1, 2), 3.0
(2, 3), 7.0
(3, 4), 18.0
(4, 5), 24.0
(5, 6), 41.0
(6, 7), 62.0
(7, 8), 87.0
(8, 9), 116.0
(9, 10), 145.0
(10, 11), 250.0

```



2. Analisis Grafik Perbandingan Nilai Asli Dan Nilai Prediksi

Berdasarkan grafik dengan nilai $a = i+1$ dan $b = i+2$ didapatkan hasil yang terlihat bahwa metode SVM menghasilkan prediksi yang sangat akurat untuk setiap fungsi. Pada grafik tersebut, sumbu x merepresentasikan indeks data, sumbu y menunjukkan variabel nilai, garis biru menggambarkan nilai asli, dan garis kuning putus-putus mewakili nilai prediksi dari SVM. Dari pola yang tampak, nilai cenderung meningkat seiring bertambahnya indeks data.

3. Kode Program Dengan Nilai $a = i+2$ dan $b = i+4$

```

55. #Kode Program SVM
56.
57. from sklearn import svm
58.
59. #Database : Gerbang Logika AND
60. #X=Data, y=target
61. x = [[0, 0], [0, 1], [1, 0], [1,1]]
62. y = [0, 0, 0, 1]
63.
64. #training and classify
65. clf = svm.SVC()
66. clf.fit(x,y)
67.
68. #Prediksi
69. print ("Logika AND Metode Support Vector Machine (SVM)")
70. print ("Logika = Prediksi")
71. print ("0 0 =",clf.predict([[0, 0]]))
72. print ("0 1 =",clf.predict([[0, 1]]))
73. print ("1 0 =",clf.predict([[1, 0]]))
74. print ("1 1 =",clf.predict([[1, 1]]))
75.
76. # Fungsi untuk mencari integral menggunakan metode Trapezoid
77. def trapezoid(a, b, f):
78.     # Fungsi Trapezoid
79.     n = 100 # Jumlah partisi
80.     h = (b - a) / n
81.     sum = 0.0
82.     for i in range(1, n):

```

```

83.     x = a + i * h
84.     sum += f(x)
85.     integral = (h / 2) * (f(a) + 2 * sum + f(b)) # Rumus Trapezoid
86.     print(a, ",", b, ",", round(integral, 2)) # Menampilkan hasil integral
87.     return integral
88.
89. # Daftar fungsi menggunakan lambda
90. functions = [
91.     lambda x: 2 * x,      # Fungsi pertama: f(x) = 2x
92.     lambda x: 2 * x + 2,  # Fungsi kedua: f(x) = 2x + 2
93.     lambda x: 4 * x + 4,  # Fungsi ketiga: f(x) = 4x + 4
94.     lambda x: 4 * x + 6,  # Fungsi keempat: f(x) = 4x + 6
95.     lambda x: 6 * x + 8,  # Fungsi kelima: f(x) = 6x + 8
96.     lambda x: 8 * x + 10, # Fungsi keenam: f(x) = 8x + 10
97.     lambda x: 10 * x + 12, # Fungsi ketujuh: f(x) = 10x + 12
98.     lambda x: 12 * x + 14, # Fungsi kedelapan: f(x) = 12x + 14
99.     lambda x: 14 * x + 12, # Fungsi kesembilan: f(x) = 14x + 12
100.    lambda x: 20 * x + 40 # Fungsi kesepuluh: f(x) = 20x + 40
101.    ]
102.
103.    # Looping untuk menghitung integral masing-masing fungsi
104.    for i, func in enumerate(functions):
105.        a = i + 2 # Batas bawah
106.        b = i + 4 # Batas atas
107.        trapezoid(a, b, func) # Menghitung integral

```

Hasil Prediksi

```

a,b,Target
2 , 4 , 12.0
3 , 5 , 20.0
4 , 6 , 48.0
5 , 7 , 60.0
6 , 8 , 100.0
7 , 9 , 148.0
8 , 10 , 204.0
9 , 11 , 268.0
10 , 12 , 332.0
11 , 13 , 560.0

```

Import library yang diperlukan

```

import numpy as np
import pandas as pd
from sklearn import svm
from google.colab import drive
import matplotlib.pyplot as plt

```

```

# Mount Google Drive
drive.mount('/content/drive')

# Path ke file Database.txt di Google Drive
file_path = '/content/drive/My Drive/Function.txt' # Ganti dengan path sesuai lokasi
file Anda di Google Drive

# Membaca data dari file
Database = pd.read_csv(file_path, sep=',', header=0)

# X = data, y = target
X = Database[['a', 'b']] # Pastikan kolom sesuai dengan nama yang ada di file
y = Database['Target']

# Membuat dan melatih model SVM
clf = svm.SVC()
clf.fit(X.values, y)

# Melakukan prediksi
y_pred = clf.predict(X.values)

# Menampilkan hasil prediksi
print("Hasil prediksi:")
for i, pred in enumerate(y_pred):
    print(f"({X.iloc[i, 0]}, {X.iloc[i, 1]}), {pred}")

# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)', color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi (SVM)',
color='yellow')

# Menambahkan label dan judul
plt.xlabel('Indeks Data')
plt.ylabel('Nilai')
plt.title('Perbandingan Nilai Asli vs Nilai Prediksi Menggunakan SVM')
plt.legend()
# Menambahkan grid
plt.grid()

plt.show()

```

Mounted at /content/drive

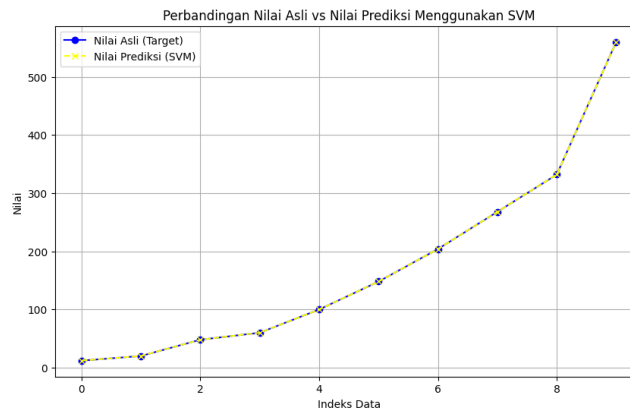
Hasil prediksi:

(2, 4), 12.0

(3, 5), 20.0

(4, 6), 48.0

(5, 7), 60.0
 (6, 8), 100.0
 (7, 9), 148.0
 (8, 10), 204.0
 (9, 11), 268.0
 (10, 12), 332.0
 (11, 13), 560.0



4. Analisis Grafik Perbandingan Nilai Asli Dan Nilai Prediksi Pada Soal Nomor 3

Berdasarkan grafik yang dihasilkan dengan nilai $a = i+2$ dan $b = i+4$, terlihat bahwa metode SVM memberikan prediksi yang sangat akurat untuk setiap fungsi. Pada grafik tersebut, sumbu x merepresentasikan indeks data, sumbu y menunjukkan variabel nilai, garis biru mewakili nilai asli, dan garis kuning putus-putus menggambarkan nilai prediksi dari SVM. Grafik tersebut menunjukkan bahwa nilai cenderung meningkat seiring bertambahnya indeks data. Selain itu, kurva yang dihasilkan pada tugas ketiga ini memiliki bentuk kurva yang sama dengan tugas pertama, karena fungsi yang digunakan pada keduanya sama akan tetapi memiliki nilai yang lebih tinggi dibandingkan dengan tugas pertama.

5. Algoritma program integral trapezoid dan prediksi nilai integral trapezoid menggunakan metode Support Vector Machine

Menghitung Prediksi Integral Trapezoid

1. Mengimpor *library* yang diperlukan, termasuk pustaka sklearn untuk implementasi *Support Vector Machine* (SVM).
2. Menyiapkan dataset berupa data gerbang logika AND, di mana x merepresentasikan data input dan y sebagai target output.
3. Membuat model SVM dengan menggunakan `svm.SVC()` dan melatihnya menggunakan data input x dan target y.
4. Dan dapatkan outputnya berupa logika
5. Mendefinisikan fungsi trapezoid dengan parameter berupa nilai batas atas (a), batas bawah (b), dan fungsi yang akan di integralkan (f).

6. Menentukan rumus trapezoid, di mana selisih antara batas bawah dan batas atas dibagi dengan jumlah partisi data.
7. Mendefinisikan fungsi yang akan dihitung untuk menghitung nilai prediksi integralnya.
8. Menyimpan output perhitungan prediksi integral dalam format file teks (.txt).

Memplot Nilai Prediksi SVM

1. Import *library* yang dibutuhkan kemudian kita juga memanggil fungsi svm dari sklearn.
2. Melakukan *mount* Google Drive untuk mengakses file .txt yang telah di upload ke my drive.
3. Mendefinisikan sumbu x sebagai nilai batas atas (*a*) dan batas bawah (*b*), serta sumbu y sebagai target prediksi.
4. Menampilkan hasil prediksi sesuai instruksi, kemudian memplot grafik yang menunjukkan perbandingan antara nilai asli dan nilai prediksi menggunakan SVM.