

PRAKTIKUM FISIKA KOMPUTASI

SUPPORT VECTOR MACHINE (SVM)

Darniel Trio Apriliansyah

NIM. 1227030009

1. Buat prediksi integral trapezoid menggunakan Support Vector Machine pada persamaan berikut ini:

Kode Program

```
def Trapezoid(a,b,f):
    n = 100
    def trapezoid(f,a,b,n=100):
        h = (b - a)/n
        sum = 0.0
        for i in range (1,n):
            x = a + i * h
            sum = sum + f(x)
        integral = (h/2)*(f(a)+2*sum+f(b))
        return integral
    integral = trapezoid(f,a,b,n)
    print(a,",",b,",",round(integral,2))

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x + 2)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 2*x + 4)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 4*x + 6)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 6*x + 8)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 8*x + 10)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 10*x + 12)
```

```

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 12*x + 14)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 14*x + 40)

for i in range(0,5):
    Trapezoid(i+1,i+2,lambda x: 20*x + 40)

```

Hasil

1	,	2	,	3.0	1	,	2	,	22.0
2	,	3	,	5.0	2	,	3	,	30.0
3	,	4	,	7.0	3	,	4	,	38.0
4	,	5	,	9.0	4	,	5	,	46.0
5	,	6	,	11.0	5	,	6	,	54.0
1	,	2	,	5.0	1	,	2	,	27.0
2	,	3	,	7.0	2	,	3	,	37.0
3	,	4	,	9.0	3	,	4	,	47.0
4	,	5	,	11.0	4	,	5	,	57.0
5	,	6	,	13.0	5	,	6	,	67.0
1	,	2	,	7.0	1	,	2	,	32.0
2	,	3	,	9.0	2	,	3	,	44.0
3	,	4	,	11.0	3	,	4	,	56.0
4	,	5	,	13.0	4	,	5	,	68.0
5	,	6	,	15.0	5	,	6	,	80.0
1	,	2	,	12.0	1	,	2	,	33.0
2	,	3	,	16.0	2	,	3	,	47.0
3	,	4	,	20.0	3	,	4	,	61.0
4	,	5	,	24.0	4	,	5	,	75.0
5	,	6	,	28.0	5	,	6	,	89.0
1	,	2	,	17.0	1	,	2	,	70.0
2	,	3	,	23.0	2	,	3	,	90.0
3	,	4	,	29.0	3	,	4	,	110.0
4	,	5	,	35.0	4	,	5	,	130.0
5	,	6	,	41.0	5	,	6	,	150.0

2. Analisis grafik perbandingan nilai asli dan nilai prediksi pada soal nomor 1!

Kode Program

```
# 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/MyDrive/Trapezoid.txt' # Ganti dengan
path sesuai lokasi file Anda di Google Drive

# Membaca data dari file (periksa delimiter, misalnya koma atau tab)
Database = pd.read_csv(file_path, sep=",", header=0) # Pastikan
delimiter sesuai dengan file

# Cek isi data untuk memastikan format benar
print(Database.head())

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

# 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))

# Periksa panjang data sebelum plotting untuk menghindari error
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()
plt.grid()
plt.show()
```

Hasil



1 , 2 , 7.0	3 , 4 , 9.0	5 , 6 , 11.0
2 , 3 , 7.0	4 , 5 , 11.0	1 , 2 , 7.0
3 , 4 , 9.0	5 , 6 , 11.0	2 , 3 , 7.0
4 , 5 , 11.0	1 , 2 , 7.0	3 , 4 , 9.0
5 , 6 , 11.0	2 , 3 , 7.0	4 , 5 , 11.0
1 , 2 , 7.0	3 , 4 , 9.0	5 , 6 , 11.0
2 , 3 , 7.0	4 , 5 , 11.0	
3 , 4 , 9.0	5 , 6 , 11.0	
4 , 5 , 11.0	1 , 2 , 7.0	
5 , 6 , 11.0	2 , 3 , 7.0	
1 , 2 , 7.0	3 , 4 , 9.0	
2 , 3 , 7.0	4 , 5 , 11.0	
3 , 4 , 9.0	5 , 6 , 11.0	
4 , 5 , 11.0	1 , 2 , 7.0	
5 , 6 , 11.0	2 , 3 , 7.0	
1 , 2 , 7.0	3 , 4 , 9.0	
2 , 3 , 7.0	4 , 5 , 11.0	

Analisis

Grafik perbandingan antara nilai asli (target) dan nilai prediksi menggunakan metode Support Vector Machine (SVM) menunjukkan perbedaan yang cukup mencolok, terutama pada bagian akhir grafik. Di awal data, prediksi SVM cukup dekat dengan nilai asli, namun pada titik sekitar indeks 30 dan seterusnya, prediksi SVM mulai menunjukkan perbedaan yang signifikan. Sementara nilai asli terus meningkat dengan tajam, prediksi SVM justru terlihat lebih datar dan tidak bisa mengikuti lonjakan yang terjadi pada nilai asli. Hal ini menunjukkan bahwa model SVM kesulitan untuk menangkap perubahan yang cepat dan besar pada data. Model ini lebih efektif pada data dengan perubahan yang lebih stabil dan bertahap, tetapi kesulitan saat harus merespons lonjakan yang tajam. Meskipun prediksi SVM dapat mengikuti pola pada sebagian besar data, ada ketidakakuratan yang cukup besar ketika nilai asli mengalami kenaikan drastis. Untuk memperbaiki hasil ini, mungkin diperlukan penyesuaian parameter model

3. Kerjakan soal nomor 1 dengan nilai $a = i+2$ dan $b = i+4$!

Kode Program

```
def Trapezoid(a,b,f):
    n = 100
    def trapezoid(f,a,b,n=100):
        h = (b - a)/n
        sum = 0.0
        for i in range (1,n):
            x = a + i * h
            sum = sum + f(x)
        integral = (h/2)*(f(a)+2*sum+f(b))
        return integral
    integral = trapezoid(f,a,b,n)
    print(a,"",b,"",round(integral,2))

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 2*x)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 2*x + 2)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 2*x + 4)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 4*x + 6)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 6*x + 8)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 8*x + 10)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 10*x + 12)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 12*x + 14)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 14*x + 40)

for i in range(0,5):
    Trapezoid(i+2,i+4,lambda x: 20*x + 40)
```

Hasil

2 , 4 , 12.0	2 , 4 , 84.0
3 , 5 , 16.0	3 , 5 , 104.0
4 , 6 , 20.0	4 , 6 , 124.0
5 , 7 , 24.0	5 , 7 , 144.0
6 , 8 , 28.0	6 , 8 , 164.0
2 , 4 , 16.0	2 , 4 , 100.0
3 , 5 , 20.0	3 , 5 , 124.0
4 , 6 , 24.0	4 , 6 , 148.0
5 , 7 , 28.0	5 , 7 , 172.0
6 , 8 , 32.0	6 , 8 , 196.0
2 , 4 , 20.0	2 , 4 , 164.0
3 , 5 , 24.0	3 , 5 , 192.0
4 , 6 , 28.0	4 , 6 , 220.0
5 , 7 , 32.0	5 , 7 , 248.0
6 , 8 , 36.0	6 , 8 , 276.0
2 , 4 , 36.0	2 , 4 , 200.0
3 , 5 , 44.0	3 , 5 , 240.0
4 , 6 , 52.0	4 , 6 , 280.0
5 , 7 , 60.0	5 , 7 , 320.0
6 , 8 , 68.0	6 , 8 , 360.0
2 , 4 , 52.0	
3 , 5 , 64.0	
4 , 6 , 76.0	
5 , 7 , 88.0	
6 , 8 , 100.0	
2 , 4 , 68.0	
3 , 5 , 84.0	
4 , 6 , 100.0	
5 , 7 , 116.0	
6 , 8 , 132.0	

4. Analisis grafik perbandingan nilai asli dan nilai prediksi pada soal nomor 3!

Kode Program

```
# 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/MyDrive/Trapezoid3.txt' # Ganti dengan
path sesuai lokasi file Anda di Google Drive

# Membaca data dari file (periksa delimiter, misalnya koma atau tab)
Database = pd.read_csv(file_path, sep=",", header=0) # Pastikan
delimiter sesuai dengan file

# Cek isi data untuk memastikan format benar
print(Database.head())

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

# 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))

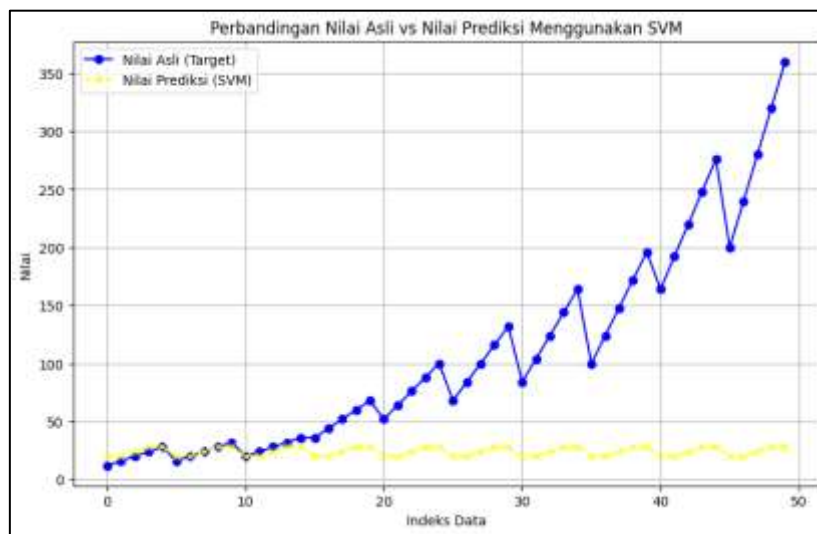
# Periksa panjang data sebelum plotting untuk menghindari error
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)',
color='blue')
```



```
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()
plt.grid()
plt.show()
```

Hasil



2	4	20.0
3	5	20.0
4	6	24.0
5	7	28.0
6	8	28.0
2	4	20.0
3	5	20.0
4	6	24.0
5	7	28.0
6	8	28.0
2	4	20.0
3	5	20.0
4	6	24.0
5	7	28.0
6	8	28.0
2	4	20.0
3	5	20.0
4	6	24.0
5	7	28.0
6	8	28.0
2	4	20.0
3	5	20.0
4	6	24.0
5	7	28.0
6	8	28.0

Analisis

Grafik yang menunjukkan perbandingan antara nilai asli (target) dan nilai prediksi dengan menggunakan metode Support Vector Machine (SVM) memperlihatkan perbedaan yang cukup besar, terutama pada bagian akhir data. Di awal grafik, prediksi SVM cukup mendekati nilai asli, tetapi setelah indeks sekitar 35, prediksi mulai tampak jauh berbeda dengan nilai asli. Pada bagian ini, nilai asli terus naik dengan tajam, sementara prediksi SVM justru lebih datar dan tidak dapat mengikuti lonjakan tersebut. Perbedaan ini menunjukkan bahwa model SVM kesulitan menangkap perubahan yang sangat cepat atau perubahan besar yang terjadi pada bagian akhir data. Hal ini mungkin terjadi karena parameter model SVM yang tidak sepenuhnya cocok untuk jenis data tersebut, atau modelnya tidak cukup fleksibel untuk merespons perubahan yang begitu tajam.

5. Jelaskan algoritma program integral trapezoid dan prediksi nilai integral trapezoid menggunakan metode Support Vector Machine!

Metode integral trapezoid adalah teknik numerik yang digunakan untuk menghitung nilai integral suatu fungsi. Cara kerjanya adalah dengan membagi area di bawah kurva fungsi menjadi beberapa trapezoid kecil, lalu menghitung luas setiap trapezoid dan menjumlahkannya untuk mendapatkan perkiraan nilai integral. Langkah pertama adalah membagi interval integrasi menjadi beberapa subinterval yang sama panjang. Kemudian, kita menghitung nilai fungsi pada titik-titik di sepanjang interval tersebut. Luas tiap trapezoid dihitung dengan rumus yang melibatkan dua nilai fungsi pada batas subinterval, yaitu pada titik awal dan titik akhir. Setelah itu, luas seluruh trapezoid dijumlahkan untuk menghasilkan nilai integral yang diperkirakan. Semakin banyak subinterval yang digunakan, semakin akurat hasilnya.

Sedangkan untuk prediksi nilai integral menggunakan metode Support Vector Machine (SVM), konsepnya sedikit berbeda. SVM, yang umumnya digunakan dalam bidang klasifikasi dan regresi, dapat dilatih untuk memprediksi nilai integral berdasarkan data yang ada. Untuk itu, pertama-tama kita mengumpulkan data yang berisi nilai-nilai fungsi pada titik-titik tertentu dalam interval integrasi, serta hasil integral yang dihitung dengan metode trapezoid. Data ini digunakan untuk melatih model SVM, di mana model akan belajar mengenali pola antara nilai-nilai titik interval dan hasil integralnya. Setelah model terlatih, kita bisa memberinya nilai titik baru, dan model akan memprediksi hasil integralnya. Dengan menggunakan SVM, kita bisa mendapatkan perkiraan nilai integral dengan cepat, terutama jika kita memiliki banyak data dan ingin menghindari perhitungan integral yang rumit.