

NAMA : NOVIANTI ZAKIAH

NIM : 1227030026

1. Prediksi integral trapezoid menggunakan Support Vector Machine

```
#Kode Program SVM

from sklearn import svm

#Database: Gerbang Logika AND
#X=Data, y=target
x = [[0, 0], [0, 1], [1, 0], [1,1]]
y = [0, 0, 0, 1]

#training and classify
clf = svm.SVC()
clf.fit(x,y)

#Prediksi
print ("Logika AND Metode Support Vector Machine (SVM)")
print ("Logika = Prediksi")
print ("0 0 =",clf.predict([[0, 0]]))
print ("0 1 =",clf.predict([[0, 1]]))
print ("1 0 =",clf.predict([[1, 0]]))
print ("1 1 =",clf.predict([[1, 1]]))

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 += f(x)
        integral = (h / 2) * (f(a) + 2 * sum + f(b))
        print(a," ",b," ",round(integral,2))
        return integral

# Melakukan looping untuk membuat database dari beberapa soal
integral
functions = [
    lambda x: 2 * x,          # Fungsi pertama: f(x) = 2x
    lambda x: 2 * x + 2,      # Fungsi kedua: f(x) = 2x + 2
```

```

lambda x: 4 * x + 4,      # Fungsi ketiga:  $f(x) = 4x + 4$ 
lambda x: 4 * x + 6,      # Fungsi keempat:  $f(x) = 4x + 6$ 
lambda x: 6 * x + 8,      # Fungsi kelima:  $f(x) = 6x + 8$ 
lambda x: 8 * x + 10,     # Fungsi keenam:  $f(x) = 8x + 10$ 
lambda x: 10 * x + 12,    # Fungsi ketujuh:  $f(x) = 10x + 12$ 
lambda x: 12 * x + 14,    # Fungsi kedelapan:  $f(x) = 12x + 14$ 
lambda x: 14 * x + 12,    # Fungsi kesembilan:  $f(x) = 14x + 12$ 
12
lambda x: 20 * x + 40     # Fungsi kesepuluh:  $f(x) = 20x + 40$ 
]

# Looping untuk menghitung integral masing-masing fungsi
for i, func in enumerate(functions):
    a = i + 1 # Batas bawah
    b = i + 2 # Batas atas
    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/pertamaa.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}")

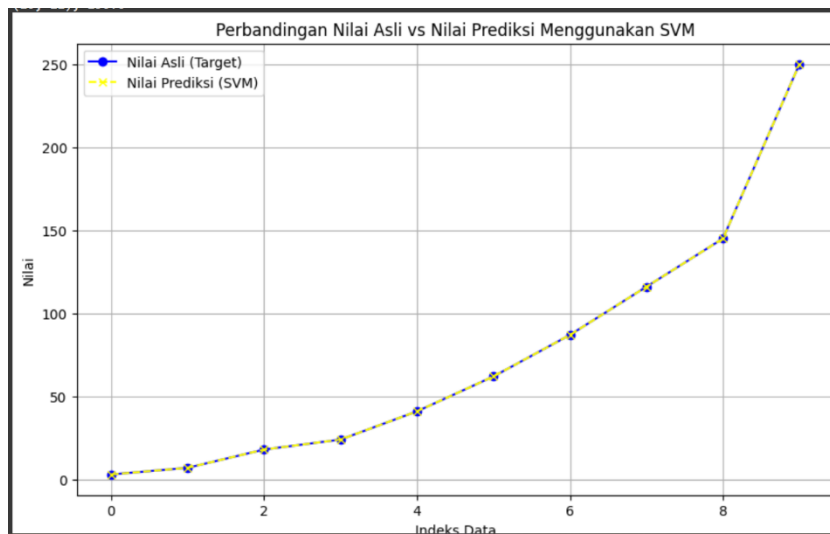
# Membuat plot perbandingan nilai asli dengan nilai prediksi
plt.figure(figsize=(10, 6))
plt.plot(range(len(y)), y, 'o-', label='Nilai Asli (Target)',
color='blue')
plt.plot(range(len(y_pred)), y_pred, 'x--', label='Nilai Prediksi
(SVM)',
color='yellow')

# 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_re
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

Grafik ini menunjukkan perbandingan antara nilai asli (target) dan nilai prediksi menggunakan SVM (Support Vector Machine). Dari grafik terlihat bahwa garis biru yang mewakili nilai asli dan garis kuning dengan tanda x untuk nilai prediksi memiliki pola yang hampir sama, menandakan bahwa model SVM mampu memprediksi nilai dengan sangat baik. Kesalahan atau selisih antara kedua garis terlihat sangat kecil, menunjukkan kesamaan atau akurasi yang tinggi. Grafik ini mengikuti pola pertumbuhan non-linear yang jelas terlihat dari bentuk kurva yang semakin meningkat signifikan pada indeks data terakhir. Model SVM berhasil menangkap tren tersebut dengan baik, di mana prediksi tetap mengikuti nilai asli di setiap titik data. Secara keseluruhan, model SVM menunjukkan performa yang baik dalam memprediksi data target dengan tingkat akurasi yang tinggi.

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

```
#Kode Program SVM

from sklearn import svm

#Database: Gerbang Logika AND
#X=Data, y=target
x = [[0, 0], [0, 1], [1, 0], [1, 1]]
```

```

y = [0, 0, 0, 1]

#training and classify
clf = svm.SVC()
clf.fit(x,y)

#Prediksi
print ("Logika AND Metode Support Vector Machine (SVM)")
print ("Logika = Prediksi")
print ("0 0 =",clf.predict([[0, 0]]))
print ("0 1 =",clf.predict([[0, 1]]))
print ("1 0 =",clf.predict([[1, 0]]))
print ("1 1 =",clf.predict([[1, 1]]))

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 += f(x)
        integral = (h / 2) * (f(a) + 2 * sum + f(b))
        print(a," ",b," ",round(integral,2))
        return integral

# Melakukan looping untuk membuat database dari beberapa soal
integral
functions = [
    lambda x: 2 * x,          # Fungsi pertama: f(x) = 2x
    lambda x: 2 * x + 2,      # Fungsi kedua: f(x) = 2x + 2
    lambda x: 4 * x + 4,      # Fungsi ketiga: f(x) = 4x + 4
    lambda x: 4 * x + 6,      # Fungsi keempat: f(x) = 4x + 6
    lambda x: 6 * x + 8,      # Fungsi kelima: f(x) = 6x + 8
    lambda x: 8 * x + 10,     # Fungsi keenam: f(x) = 8x + 10
    lambda x: 10 * x + 12,    # Fungsi ketujuh: f(x) = 10x + 12
    lambda x: 12 * x + 14,    # Fungsi kedelapan: f(x) = 12x + 14
    lambda x: 14 * x + 12,    # Fungsi kesembilan: f(x) = 14x +
12
    lambda x: 20 * x + 40     # Fungsi kesepuluh: f(x) = 20x + 40
]

# Looping untuk menghitung integral masing-masing fungsi
for i, func in enumerate(functions):

```

```

a = i + 2 # Batas bawah
b = i + 4 # Batas atas
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]
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 Function.txt di Google Drive
file_path='/content/drive/My Drive/function.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()

```

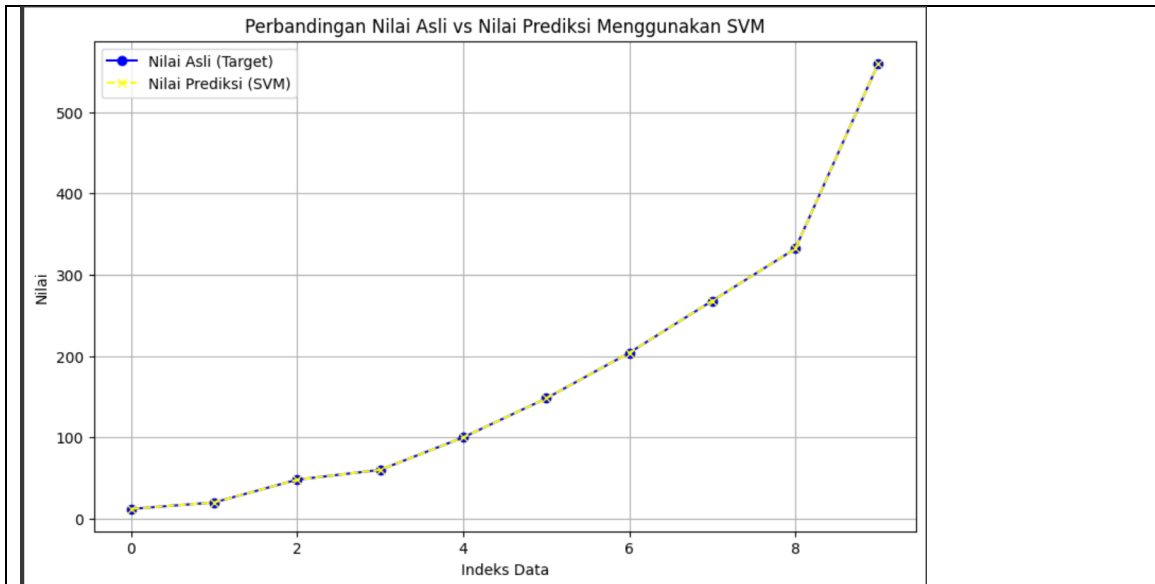
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

Hasil prediksi:

```

(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

Grafik ini membandingkan nilai asli (target) dengan nilai prediksi menggunakan SVM (Support Vector Machine). Dari visualisasi ini, terlihat bahwa garis biru yang mewakili nilai asli (target) dan garis kuning (dengan tanda silang) untuk nilai prediksi memiliki pola yang hampir sama di setiap titik data. Hal ini menunjukkan bahwa model SVM mampu memprediksi nilai dengan sangat baik, karena perbedaan atau kesalahan antara nilai asli dan nilai prediksi hampir tidak terlihat. Secara khusus, grafik menunjukkan tren non-linear yang cukup signifikan, di mana nilai meningkat secara bertahap pada indeks data awal, lalu naik semakin pesat menuju indeks data terakhir. Model SVM terlihat berhasil menangkap tren peningkatan tersebut dengan akurasi yang tinggi. Pola garis prediksi yang mengikuti nilai asli menunjukkan bahwa SVM memiliki kemampuan yang baik dalam mempelajari hubungan antara data input dan output. Dengan demikian, performa prediksi dari model SVM pada grafik ini dapat dikatakan akurat.

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

1. Upload library yang dibutuhkan seperti "from sklearn import svm"
2. Masukkan database gerbang logika AND dengan nilai x input dan y output
3. Masukkan data prediksi
4. Tambahkan kode untuk perhitungan atau persamaan trapezoid
5. Tambahkan juga fungsi integral dan nilai a,b (sebagai batas bawah dan atas)
6. Diperoleh nilai data logika = prediksi
7. Masukkan nilai logika=prediksi ke file bentuk txt untuk perbandingan grafik
8. Masukkan library untuk memunculkan grafik perbandingan antara nilai asli dan nilai prediksi
9. Masukkan kode untuk mengakses atau memanggil data yang dalam bentuk txt
10. Diperoleh hasil prediksi dan grafik perbandingan nilai asli dan nilai prediksi