

NAMA : Sherly Mawarni Kusumah

NIM : 1227030033

PRAKTIKUM FISIKA KOMPUTASI

“DECISION TREE NEIGHBORS”

- 1. Klasifikasi soal di bawah ini dengan menggunakan metode Decision Tree dengan dataset sebagai berikut.**

```
x = [[0 , 0, 0],  
      [0 , 5, 0],  
      [0 , 0, 5],  
      [0 , 5, 5],  
      [5 , 5, 0],  
      [5 , 0, 5],  
      [5 , 5, 5],  
      [10, 5, 5],  
      [5 , 10, 5],  
      [10, 10, 10]  
      ]  
y = [0,0,0,5,5,5,10,10,5,0]
```

Kemudian prediksi data berikut:

```
Logika = Prediksi  
10 10 5 =  
5 10 2 =  
2 0 10 =  
5 0 2 =  
0 0 2 =  
2 10 2 =  
1 12 5 =  
2 2 6 =  
10 5 7 =
```

- 2. Buatlah prediksi data cosinus di bawah dengan menggunakan metode Decision Tree! Ingat sebelum melakukan prediksi buat file data menggunakan format .txt kemudian di upload pada drive pribadi.**

Feature,Target
1,0.540302305
2,-0.416146837
3,-0.989992497
4,-0.653643621
5,0.283662185
6,0.960170287
7,0.753902254
8,-0.145500034
9,-0.911130261
10,-0.839071529
11,0.004425698
12,0.843853959
13,0.907446781
14,0.136737218
15,-0.759687913
16,-0.957659481
17,-0.275163338
18,0.660316708
19,0.988704619
20,0.408082062

JAWAB :

KODE PROGRAMNYA MENJADI

<pre>from sklearn import tree # Database Gerbang Logika AND # X = Data,y=Target x = [[0, 0, 0], [0, 5, 0], [0, 0, 5], [0, 5, 5], [5, 5, 0], [5, 0, 5], [5, 5, 5], [10, 5, 5], [5, 10, 5], [10, 10, 10]] y = [0,0,0,5,5,5,10,10,5,0] # Training and Classify clf = tree.DecisionTreeClassifier() clf = clf.fit(x,y)</pre>
--

```
# Prediction
print("Logika AND Metode Decision Tree")
print("Logika = Prediksi")
print("10 10 5 = ", clf.predict([[10, 10, 5]]))
print("5 10 2 = ", clf.predict([[5, 10, 2]]))
print("2 0 10 = ", clf.predict([[2, 0, 10]]))
print("5 0 2 = ", clf.predict([[5, 0, 2]]))
print("0 0 2 = ", clf.predict([[0, 0, 2]]))
print("2 10 2 = ", clf.predict([[2, 10, 2]]))
print("1 12 5 = ", clf.predict([[1, 12, 5]]))
print("2 2 6 = ", clf.predict([[2, 2, 6]]))
print("10 5 7 = ", clf.predict([[10, 5, 7]]))
```

Logika AND Metode Decision Tree

Logika = Prediksi

10 10 5 = [10]

5 10 2 = [5]

2 0 10 = [0]

5 0 2 = [5]

0 0 2 = [0]

2 10 2 = [0]

1 12 5 = [5]

2 2 6 = [0]

10 5 7 = [10]

```
from google.colab import drive
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor
import matplotlib.pyplot as plt
```

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

#Path ke file di Google Drive

FileDB = '/content/drive/My Drive/MODUL 9 PRAKFIKOM/Cosinus.txt' # Sesuai path file

Database = pd.read_csv(FileDB,sep=";",header=0)

#Lihat data

```
print("-----")
```

```
print(Database)
```

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

```
-----
  Feature  Target
0        1  0.540302
1        2 -0.416147
2        3 -0.989992
3        4 -0.653644
4        5  0.283662
5        6  0.960170
6        7  0.753902
7        8 -0.145500
```

```

8      9 -0.911130
9     10 -0.839072
10    11  0.004426
11    12  0.843854
12    13  0.907447
13    14  0.136737
14    15 -0.759688
15    16 -0.957659
16    17 -0.275163
17    18  0.660317
18    19  0.988705
19    20  0.408082

```

```

# x data, y target
x = Database[['Feature']]
y = Database.Target

```

```

reg = DecisionTreeRegressor(random_state=1)
reg = reg.fit(x,y)

```

```

# Display predicted data
xx = np.arange(1, 21, 1)
n = len(xx)
print("xx(i) Decision Tree")
for i in range(n):
    y_dct = reg.predict([[xx[i]]])
    print('{:.2f}'.format(xx[i]), y_dct)

```

```

# Plot the predicted data
y_dct2 = reg.predict(x)
plt.figure()
plt.plot(x, y_dct2, color='red')
plt.scatter(x, y, color='blue')
plt.title('Prediksi Data Menggunakan Decision Tree')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(['Decision Tree', 'data'], loc=2)
plt.show()

```

```

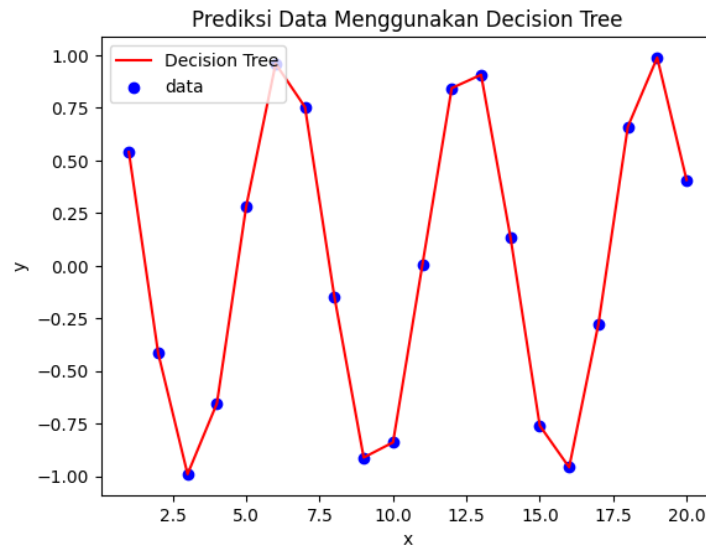
xx(i) Decision Tree
1.00 [0.5403023]
2.00 [-0.41614684]
3.00 [-0.9899925]
4.00 [-0.65364362]
5.00 [0.28366219]
6.00 [0.96017029]
7.00 [0.75390225]
8.00 [-0.14550003]
9.00 [-0.91113026]
10.00 [-0.83907153]
11.00 [0.0044257]
12.00 [0.84385396]
13.00 [0.90744678]
14.00 [0.13673722]

```

15.00	[-0.75968791]
16.00	[-0.95765948]
17.00	[-0.27516334]
18.00	[0.66031671]
19.00	[0.98870462]
20.00	[0.40808206]

3. Jelaskan hasil dari setiap metode yang telah dikerjakan dengan bahasa sendiri!

JAWAB :



Grafik di atas terdapat titik-titik biru yang menggambarkan data aktual, sedangkan garis merah menunjukkan hasil prediksi model. Decision Tree memprediksi data dengan mempartisi ruang menjadi segmen-segmen tertentu, menghasilkan prediksi berbentuk potongan (piecewise constant) yang terlihat tajam dan tidak mulus. Metode ini efektif untuk menangkap pola data yang tidak linier, namun sering kali menunjukkan keterbatasan jika model terlalu sederhana atau terlalu kompleks (overfitting). Garis prediksi cenderung mengikuti data aktual dengan cukup baik, tetapi masih ada beberapa perbedaan pada titik tertentu.

4. Menurutmu metode ini bisa digunakan untuk apa saja di dunia perkuliahan terutama di jurusan Fisika? Sebutkan minimal 3 penggunaan metode ini di perkuliahan Fisika!

JAWAB :

- ☐ **Klasifikasi Material Berdasarkan Sifat Fisikanya:** Decision Tree dapat digunakan untuk mengklasifikasikan material (misalnya logam, semikonduktor, isolator) berdasarkan parameter seperti konduktivitas listrik, massa jenis, dan kemampuan menyerap panas. Hal ini berguna dalam studi Fisika Material.
- ☐ **Prediksi Fenomena Cuaca atau Iklim Fisis:** Dalam Fisika atmosfer, metode ini dapat memprediksi fenomena seperti pola hujan, suhu, atau tekanan atmosfer berdasarkan data historis. Ini relevan untuk penelitian tentang perubahan iklim atau dinamika atmosfer.
- ☐ **Pemodelan Dinamika Sistem Mekanika:** Dalam fisika mekanika, metode ini dapat membantu memprediksi pola gerak atau kondisi sistem berdasarkan parameter awal seperti gaya, massa, dan kecepatan. Hal ini bisa diaplikasikan dalam simulasi dinamika benda atau sistem multi-body.