

Nama : Ulfah Hasanah
NIM : 1227030036

MODUL 9 (DECISION TREE NEIGHBORS)

PEMOGRAMAN:

```
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 Calssify
```

```
clf = tree.DecisionTreeClassifier()
```

```
clf = clf.fit(x,y)
```

```
#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]]))
```

Nama : Ulfah Hasanah
NIM : 1227030036

MODUL 9 (DECISION TREE NEIGHBORS)

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/Fiskom/Ulfah_Cosinos.txt' # Sesuaikan path file
```

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

#Lihat data

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

```
print(Database)
```

Nama : Ulfah Hasanah
NIM : 1227030036

MODUL 9 (DECISION TREE NEIGHBORS)

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 predict data  
y_dct2 = reg.predict(x)  
plt.figure()
```


Nama : Ulfah Hasanah
NIM : 1227030036

MODUL 9 (DECISION TREE NEIGHBORS)

Jawaban Tugas:

Dataset yang digunakan:

```
#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]
```

- X adalah data input (fitur), dengan 3 dimensi
- Y adalah target atau label keluaran
 - Hasil Metode 1

```
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]
```

Proses Pembagian Pohon Keputusan

Decision Tree bekerja berdasarkan aturan pembagian dataset yang dibuat dari nilai fitur selama proses pelatihan. Prosesnya berupa membagi dataset terlebih dahulu berdasarkan fitur tertentu untuk mengurangi impuritas pada setiap cabang. Kemudian membuat cabang dan simpul

MODUL 9 (DECISION TREE NEIGHBORS)

daun yang mengelompokkan data dengan target serupa. Setelah pohon terbentuk, input baru dipetakan ke jalur dalam pohon keputusan sesuai dengan nilai fiturnya. Hasil akhirnya adalah simpul daun yang paling sesuai berdasarkan aturan pohon tersebut.

Aanlisis Input/Logika

- Ketika pohon mencoba memprediksi input [10, 10, 5]. Pada fitur pertama bernilai 10 cocok dengan data [10, 5, 5] dengan target 10 dan [10, 10, 10] dengan target 0. Lalu fitur kedua nilai 10 pada fitur kedua berbeda dari [10, 5, 5], tetapi mungkin tidak terlalu berpengaruh jika pohon memprioritaskan fitur pertama untuk pembagian awal. Dan fitur ketiga yaitu nilai 5 sama dengan [10, 5, 5] dengan target 10, sehingga pohon cenderung memutuskan input ini mendekati grup dengan target 10. Dengan demikian pohon kemungkinan besar memutuskan hasil berdasarkan fitur pertama (10) dan fitur ketiga (5), tanpa mempertimbangkan fitur kedua (10). Jika pembagian terbaik di pohon terjadi pada fitur pertama atau ketiga, nilai fitur kedua tidak akan digunakan.
- Selanjutnya ketika pohon mencoba memprediksi input [5, 10, 2]. Pada fitur pertama bernilai 5 cocok dengan data [5, 0, 5], [5,0,5], [5,5,5], [5,10,5]. Fitur kedua bernilai 10 paling dekat dengan data [5, 10, 5]. Input [5, 10, 2] memiliki fitur ketiga 2. Untuk data/logika [5, 10, 5] memiliki fitur ketiga 5, sedangkan [5, 0, 5] memiliki fitur ketiga 0. Meskipun fitur ketiga berbeda, fitur pertama dan kedua lebih dominan dalam pembagian pohon. Dengan demikian input [5, 10, 2] paling dekat dengan dataset [5, 10, 5] sehingga hasil prediksinya 5.
- Ketika input [2, 0, 10] diberikan untuk prediksi, fitur pertama dari input [2, 0, 10] adalah 2. Ketika pohon keputusan membangun model, diputuskan pembagian berdasarkan fitur pertama. Dalam dataset, nilai fitur pertama yang lebih kecil (misalnya 0 atau 5) lebih sering berhubungan dengan target 0, sementara fitur pertama yang lebih besar (10) berhubungan dengan target lainnya. Fitur kedua dari input adalah 0, seperti [0, 0, 0] dan [0, 0, 5]. Hal ini memperkuat prediksi bahwa input [2, 0, 10] masuk ke grup dengan target 0. Selanjutnya fitur ketiga adalah 10, meskipun nilai fitur ketiga ini cukup tinggi, pembagian pertama pada pohon keputusan lebih mungkin dipengaruhi oleh fitur pertama dan kedua yang sudah sesuai dengan data yang memiliki target 0. Dengan kata lain, fitur ketiga tidak cukup

MODUL 9 (DECISION TREE NEIGHBORS)

memengaruhi prediksi ini, dan pohon sudah memutuskan berdasarkan fitur pertama dan kedua.

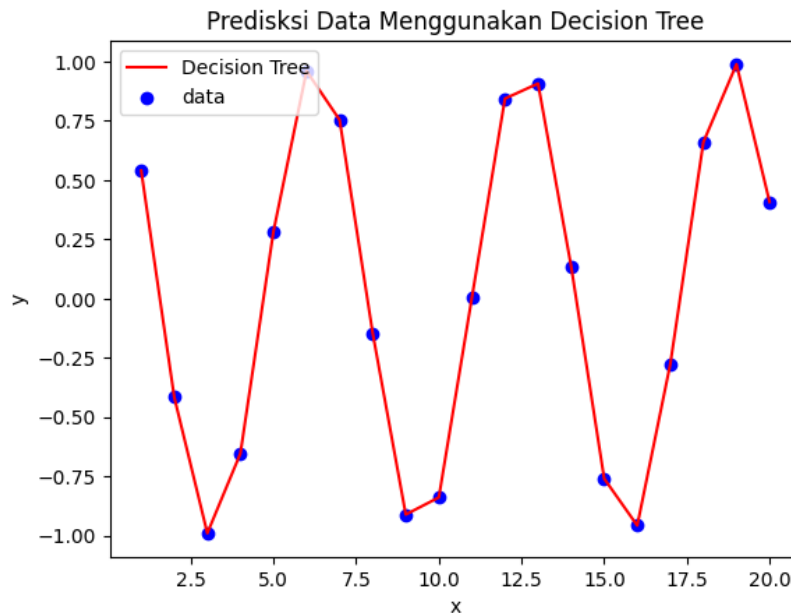
- Ketika input [5,0,2] diberikan untuk prediksi lalu akan dilakukan pencocokan dengan dibandingkan dengan pola dataset input. Lalu logika dataset input yang lebih mirip dengan logika [5, 0, 2] adalah data [5, 0, 5] dengan targetnya bernilai 5. Alasannya karena pada fitur pertama dan kedua itu sama-sama bernilai 5 dan 0. Meskipun fitur ketiganya berbeda dengan dataset, akan tetapi pohon memutuskan berdasarkan fitur pertama dan kedua. Dengan demikian input [5, 0, 2] itu menghasilkan prediksi 5.
- Ketika input [0, 0, 2] diberikan untuk prediksi. Kemudian pohon keputusan membangun model, diputuskan pembagian berdasarkan fitur pertama. Dalam dataset, fitur pertama dan kedua dari input adalah 0, seperti [0, 0, 0] dan [0, 0, 5]. Selanjutnya fitur ketiga yang lebih kecil (misalnya 0 atau 5) lebih sering berhubungan dengan target 0, sementara fitur pertama yang lebih besar (10) berhubungan dengan target lainnya. Fitur kedua dari input adalah 0, seperti [0, 0, 0] dan [0, 0, 5]. Hal ini memperkuat prediksi bahwa input [2, 0, 10] masuk ke grup dengan target 0.
- Ketika input [2, 10, 2] diberikan untuk prediksi, kemudian pohon keputusan membangun model. Dalam dataset, fitur pertama (2) lebih kecil dan lebih dekat dengan nilai data yang memiliki target 0, seperti [0, 0, 0] dan [0, 5, 0] yang memiliki target 0. Fitur kedua (10) adalah nilai yang lebih besar, namun perbedaannya tidak terlalu memengaruhi prediksi karena permbagian pertama pada pohon lebih mengutamakan fitur pertama (2). Pohon Keputusan kemungkinan besar telah menetapkan input [2, 10, 2] ke simpul yang memiliki target 0.
- Ketika input [10, 5, 7] diberikan untuk prediksi lalu akan dilakukan pencocokan dengan dibandingkan dengan pola dataset input. Lalu logika dataset input yang lebih mirip dengan logika [10, 5, 7] adalah data [10, 5, 5] dengan targetnya 10. Alasannya karena pada fitur pertama dan kedua itu sama-sama bernilai 10 dan 5. Meskipun fitur ketiganya berbeda dengan dataset, akan tetapi pohon memutuskan berdasarkan fitur pertama dan kedua. Dengan demikian input [10, 5, 7] itu menghasilkan prediksi 10.

MODUL 9 (DECISION TREE NEIGHBORS)

Datasheet yang digunakan:

	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

➤ Hasil metode 2



Grafik yang dihasilkan mencerminkan cara kerja Decision Tree Regressor dalam mempelajari pola data dan menghasilkan prediksi. Pada grafik tersebut, titik-titik biru

MODUL 9 (DECISION TREE NEIGHBORS)

menunjukkan data asli (ground truth), sedangkan garis merah mewakili hasil prediksi model. Decision Tree bekerja dengan memecah data input ke dalam simpul-simpul berdasarkan fitur (dalam hal ini kolom Feature). Model menentukan titik pemisah terbaik pada data latih untuk meminimalkan kesalahan prediksi, biasanya dengan menggunakan metrik seperti *mean squared error*. Setelah data dipisahkan, setiap kelompok data dalam simpul daun akan memiliki nilai target yang sama, yang dihitung sebagai rata-rata nilai target data di dalam simpul tersebut.

❖ 2 Metode diatas bisa digunakan di dunia perkuliahan terutama di jurusan Fisika, seperti:

1. Dalam percobaan fisika partikel, seperti yang dilakukan di laboratorium akselerator, sering kali perlu mengklasifikasikan partikel berdasarkan data karakteristik seperti energi, momentum, dan lintasan partikel. Decision Tree Classifier dapat digunakan untuk memetakan data observasi ke jenis partikel tertentu, seperti elektron, proton, atau neutron. Hal ini sangat berguna untuk mempercepat analisis data eksperimen yang besar.

2. Prediksi Parameter Eksperimen Fisika

Dalam fisika eksperimen, hubungan antara parameter input (seperti suhu, tekanan, atau medan magnet) dan hasil (seperti energi, kecepatan, atau sifat material) seringkali tidak linier dan kompleks. Decision Tree Regressor adalah metode yang tepat untuk mempelajari hubungan tersebut karena dapat bekerja dengan data non-linier dan menemukan pola dari dataset eksperimen.

3. Analisis Data Astrofisika

Bidang astrofisika sering melibatkan dataset besar dengan banyak variabel yang saling terkait, seperti pengamatan dari teleskop atau data dari simulasi astrofisika. Decision Tree dapat digunakan untuk mengidentifikasi hubungan ini dan memprediksi karakteristik fisik, seperti massa bintang, usia, atau jaraknya dari Bumi. Aplikasi ini sering digunakan dalam penelitian berbasis data besar (big data) yang umum di astrofisika.

4. Pemodelan Sistem Dinamis

Sistem dinamis dalam fisika, seperti osilasi atau gerak benda, biasanya dianalisis melalui persamaan matematis. Namun, jika data eksperimen tersedia, Decision Tree

Nama : Ulfah Hasanah
NIM : 1227030036

MODUL 9 (*DECISION TREE NEIGHBORS*)

Regressor dapat digunakan untuk mempelajari hubungan input-output tanpa perlu menyelesaikan persamaan diferensial, yang membantu dalam memahami sistem kompleks dengan lebih cepat, terutama di tingkat perkuliahan.