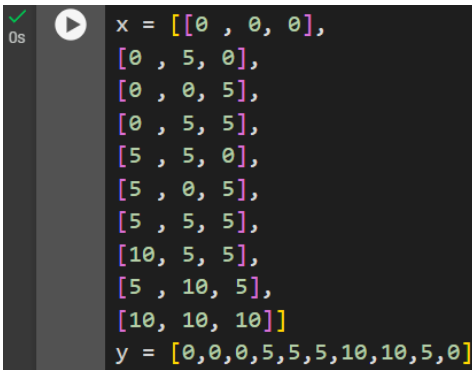



Adinda Salsabila
1227030003
Modul 9

1. 

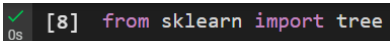
```
x = [[0, 0, 0],
      [0, 5, 0],
      [0, 0, 5],
      [0, 5, 5],
      [5, 5, 0],
      [5, 0, 5],
      [5, 5, 5],
      [10, 5, 5],
      [5, 10, 5],
      [10, 10, 10]]
y = [0, 0, 0, 5, 5, 5, 10, 10, 5, 0]
```

2. 

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

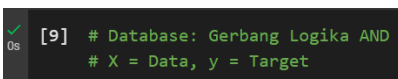
	Feature	Target
0	1	0.540302
1	2	-0.416147
2	3	-0.989992
3	4	-0.653644
4	5	0.283662
5	6	0.960170
6	7	0.753902
7	8	-0.145500
8	9	-0.911130
9	10	-0.839072
10	11	0.004426
11	12	0.843854
12	13	0.907447
13	14	0.136737
14	15	-0.759688
15	16	-0.957659
16	17	-0.275163
17	18	0.660317
18	19	0.988705
19	20	0.408082

3. Penjelasan kode program

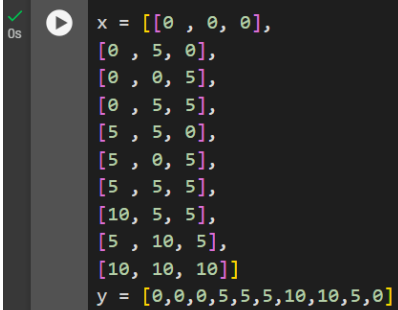


```
[8] from sklearn import tree
```

Pada bagian ini yakni mengimpor modul tree dari sklearn. Biasanya modul ini digunakan untuk pembelajaran berbasis decision tree.



```
[9] # 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]
```

Pada bagian x itu sebagai input data berupa 3 kolom. Pada bagian y itu sebagai target atau nilai yang ingin diprediksi.

```
0s [11] #Training and Classify
      clf = tree.DecisionTreeClassifier()
      clf = clf.fit(x,y)
```

Pada bagian ini yakni membuat model decision tree untuk tugas klarfikasi. Clf yakni decision tree classifier. Fit(x,y) berfungsi sebagai melatih model dengan data x dan target 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("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]
0 0 2 = [0]
2 10 2 = [0]
1 12 5 = [5]
2 2 6 = [0]
10 5 7 = [10]
```

Pada bagian ini berfungsi untuk menguji model dengan data baru. Model clf.predict memprediksi hasil logika AND dari input ("10 10 5") sampai ("10 5 7"). Output akan dicetak ke layar.

```
0s [14] from google.colab import drive
      import pandas as pd
      import numpy as np
      from sklearn.tree import DecisionTreeRegressor
      import matplotlib.pyplot as plt
```

Pada bagian pertama, berfungsi sebagai menghubungkan google drive. Pada bagian kedua berfungsi untuk mengolah data dengan pandas. Bagian ketiga berfungsi untuk membuat data acak atau memproses data numerik. Pada bagian keempat berfungsi untuk memprediksi nilai output numerik dengan memecah data. Dan pada bagian kelima berfungsi untuk memvisualisasikan data.

```
1s ▶ # x data, y target
      x = Database[['Feature']]
      y = Database[['Feature']]
```

Pada bagian x data input untuk memasukkan model yang terdiri dari nilai-nilai dalam kolom. Pada bagian y yakni targer yang ingin dipelajari pada model.

```
2s ▶ # Mount Google Drive
      drive.mount('/content/drive')

      # Path ke file di Google Drive
      FileDB = '/content/drive/My Drive/asarusabiraaa/nasdukgoceng.txt' # Sesuaikan path file
      Database = pd.read_csv(FileDB, sep=",", header=0)

      # Lihat data
      print("-----")
      print(Database)
```

Pada bagian ini menghubungkan dari google colab ke google drive untuk mengakses file. File tersebut kita masukan ke "My Drive" lalu diberi nama di filenya dan dimasukkan "Feature dan

Target” yang sudah disediakan di modul dengan mengganti nama tersebut menjadi “nasdukgoceng.txt”. Setelah di mount, file di google drive dapat digunakan di colab.

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

Pada bagian ini digunakan untuk membuat model decision tree khusus untuk tugas memprediksi nilai numerik. Model reg siap digunakan untuk melakukan prediksi terhadap data baru.

```
# Display predicted data
xx = np.arange(1, 21, 1).reshape(-1, 1) # Ubah ke array 2D -- This line created the issue
# The model expects 3 features, so create a dataset with 3 columns
xx = np.random.rand(20, 3) # Creating an array with 20 rows and 3 columns with random values
y_dct = reg.predict(xx)

print("xx(i) Decision Tree")
# Note: Since xx has multiple columns, we'll print the first element of each row
for i in range(len(xx)):
    print(f'xx[i][0]:.2f) = {y_dct[i]:.2f}')
```

```
[53] xx(i) Decision Tree
0.40 = 0.00
0.24 = 0.00
0.00 = 0.00
0.54 = 0.00
0.79 = 0.00
0.41 = 0.00
0.51 = 0.00
0.56 = 0.00
0.00 = 0.00
0.91 = 0.00
0.53 = 0.00
0.27 = 0.00
0.67 = 0.00
0.40 = 0.00
0.02 = 0.00
0.89 = 0.00
0.45 = 0.00
0.30 = 0.00
0.82 = 0.00
0.20 = 0.00
```

Pada bagian ini digunakan untuk menghasilkan data baru secara acak, memprediksi nilai keluaran dengan model memprediksi nilai numerik, dan menampilkan hasil prediksi dalam format yang rapih.

```
# ipython-input-50-6b6e803eb4e3
# Plot the predicted data
# y_dct2 = reg.predict(x) # x has only 1 feature, but the model expects 3
# Instead of x, use xx (which has 3 features) to generate predictions for the plot
y_dct2 = reg.predict(xx)
plt.figure()
plt.plot(xx[:, 0], y_dct2, color='red') # Plot against the first column of xx
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()
```

Pada bagian ini membuat grafik perbandingan antara data prediksi dan data asli. Bagian plt.plot menampilkan prediksi sebagai garis merah. Dan pada bagian plt.scatter menampilkan data asli sebagai titik biru.

4. A. Hukum gas ideal
B. Konduktivitas termal
C. Kekuatan mekanik
D. Probabilitas suatu partikel berada pada posisi tertentu dalam ruang/ keadaan eneergi tertentu