

## ▼ Import Library yang digunakan

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, r2_score, mean_squared_error
from sklearn.linear_model import LinearRegression
```

## ▼ Load Data

```
from google.colab import drive
drive.mount('/content/drive')

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

path = '/content/drive/MyDrive/PRAKTIKUM7/DATA/'
```

Perintah di atas untuk membuat alamat folder tersebut ke dalam variabel bernama path

```
df = pd.read_csv(path + "dataset_satelit.csv")
df
```

No	Longitude	Lattitude	N	P	K	Ca	Mg	Fe	Mn	...	b1	Sigma_VV	Sigma_VH	plia	lia	
0	1	103.036658	-0.604417	2.64	0.15	0.415	0.51	0.31	119.96	463.23	...	0.0433	0.18183	0.04461	35.74446	35.7974
1	2	103.037201	-0.604689	2.75	0.17	0.568	0.76	0.58	102.63	493.81	...	0.0465	0.22079	0.04640	35.12096	35.1459
2	3	103.036359	-0.603012	1.77	0.12	0.339	0.49	0.6	107.37	460.93	...	0.0417	0.18926	0.03992	35.07724	35.07730
3	4	103.036950	-0.603219	2.30	0.15	0.460	0.74	0.67	96.02	338.17	...	0.0367	0.14769	0.03622	36.08078	36.08469
4	5	103.036802	-0.601969	2.05	0.14	0.308	0.64	0.72	87.01	384.33	...	0.0361	0.18205	0.03797	32.68855	32.6929
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
589	590	103.605867	0.057633	2.49	0.16	0.347	0.78	0.86	63.38	269.95	...	0.2336	0.13050	0.09390	0.12700	0.09860
590	591	103.606717	0.057100	2.74	0.15	0.466	0.73	0.5	51.04	683.42	...	0.2506	0.21280	0.15920	0.20060	0.14730
591	592	103.606250	0.056767	2.63	0.15	0.422	0.82	0.59	82.57	396.18	...	0.3413	0.27730	0.17820	0.25790	0.18690
592	593	103.606400	0.056517	2.75	0.17	0.502	0.69	0.53	102.07	246.35	...	0.3413	0.32740	0.28760	0.29970	0.22250
593	594	103.606033	0.056317	2.71	0.15	0.419	0.78	0.46	90.60	371.46	...	0.3482	0.41280	0.36380	0.38660	0.28990

594 rows × 34 columns

Perintah di atas untuk membaca file dengan format csv menggunakan library pandas

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594 entries, 0 to 593
Data columns (total 34 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   No          594 non-null    int64  
 1   Longitude   594 non-null    float64 
 2   Lattitude   594 non-null    float64 
 3   N            594 non-null    float64 
 4   P            594 non-null    float64 
 5   K            593 non-null    float64 
 6   Ca           594 non-null    float64 
 7   Mg           594 non-null    object  
 8   Fe           594 non-null    float64 
 9   Mn           594 non-null    float64 
 10  Cu           594 non-null    float64 
 11  Zn           594 non-null    float64 
 12  B             594 non-null    float64 
 13  b1           594 non-null    float64
```

```

14 b11      594 non-null   float64
15 b9       594 non-null   float64
16 b8a     594 non-null   float64
17 b8       594 non-null   float64
18 b7       594 non-null   float64
19 b6       594 non-null   float64
20 b5       594 non-null   float64
21 b4       594 non-null   float64
22 b3       594 non-null   float64
23 b2       594 non-null   float64
24 b1       594 non-null   float64
25 Sigma_VV 594 non-null   float64
26 Sigma_VH 594 non-null   float64
27 plia     594 non-null   float64
28 lia      594 non-null   float64
29 iafe     594 non-null   float64
30 gamma0_vv 594 non-null   float64
31 gamma0_vh 594 non-null   float64
32 beta0_vv  594 non-null   float64
33 beta0_vh  594 non-null   float64
dtypes: float64(32), int64(1), object(1)
memory usage: 157.9+ KB

```

Perintah diatas untuk mengecek info dari dataset yang dibaca

- **No** : nomor urut data.
- **Longitude** : posisi garis bujur (arah timur & barat).
- **Lattitude** : posisi garis lintang (arah utara & selatan).
- **N, P, K**: kadar unsur hara utama tanah: Nitrogen, Fosfor, Kalium.
- **Ca, Mg, Fe, Mn, Cu, Zn, B, Mo, Na** : unsur kimia tanah lainnya (kalsium, magnesium, besi, mangan, tembaga, seng, boron, molibdenum, natrium).
- **pH\_H2O** : tingkat keasaman tanah diukur dengan air.
- **pH\_KCl** : tingkat keasaman tanah diukur dengan larutan KCl.
- **C\_Org** : kandungan karbon organik tanah.
- **CEC** : kapasitas tukar kation, kemampuan tanah menahan unsur hara.
- **b1 – b7** : nilai reflektansi dari band citra satelit (misal Landsat/Sentinel).
- **Sigma\_VV, Sigma\_VH** : nilai pantulan radar satelit dengan polarisasi VV dan VH.
- **plia, lia, iafe** : parameter sudut datang atau pantulan sinyal radar. keyboard\_arrow\_down Interpretasi setiap kolom
- **gamma0\_vv, gamma0\_vh, beta0\_vv, beta0\_vh** : hasil koreksi nilai pantulan radar (untuk analisis permukaan tanah).

```
df.isnull().sum()
```

```
0
No      0
Longitude 0
Latitude 0
N       0
P       0
K       1
Ca      0
Mg      0
Fe      0
Mn      0
Cu      0
Zn      0
B       0
b12     0
b11     0
b9      0
b8a    0
b8     0
b7     0
b6     0
b5     0
b4     0
b3     0
b2     0
b1     0
Sigma_VV 0
Sigma_VH 0
plia    0
lia     0
iafe    0
gamma0_vv 0
gamma0_vh 0
beta0_vv 0
beta0_vh 0
```

dtype: int64

Perintah diatas untuk mengecek missing value yang ada di dataset

```
df.describe()
```

	No	Longitude	Lattitude	N	P	K	Ca	Fe	Mn	Cu
<b>count</b>	594.000000	594.000000	594.000000	594.000000	594.000000	593.000000	594.000000	594.000000	594.000000	594.000000
<b>mean</b>	297.500000	106.878644	-1.024933	2.259091	0.141380	0.582175	0.595094	74.613771	308.034697	2.391195
<b>std</b>	171.617307	4.949840	0.965349	0.395499	0.019782	0.222567	0.366118	55.579655	241.731643	1.580296
<b>min</b>	1.000000	102.760857	-2.333750	1.140000	0.090000	0.122000	0.050000	21.080000	3.160000	0.090000
<b>25%</b>	149.250000	102.927811	-2.233338	1.982500	0.130000	0.429000	0.320000	40.705000	124.015000	1.172500
<b>50%</b>	297.500000	103.581969	-0.602276	2.280000	0.140000	0.549000	0.540000	65.650000	239.445000	2.225000
<b>75%</b>	445.750000	113.403797	-0.257349	2.570000	0.150000	0.710000	0.790000	87.372500	434.990000	3.357500
<b>max</b>	594.000000	113.434700	0.069251	3.230000	0.220000	1.489000	2.820000	559.100000	2009.320000	8.170000

8 rows × 33 columns

Perintah di atas untuk menampilkan statistik deskriptif dari kolom-kolom yang bertipe angka (numerik) dalam dataset

- **count** : jumlah data (baris) yang ada di tiap kolom
- **mean** : nilai rata-rata
- **std** : standar deviasi (tingkat sebaran data dari rata-ratanya)
- **min** : nilai terkecil
- **25%** : nilai pada persentil 25 (seperempat bagian bawah data)
- **50%** : nilai tengah (median)
- **75%**: nilai pada persentil 75 (tiga perempat bagian atas data)
- **max** : nilai terbesar

df.duplicated().sum()

np.int64(0)

df = df.drop\_duplicates()

df.duplicated().sum()

np.int64(0)

Perintah diatas untuk mengecek data duplikat yang ada didataset dan mengdropnya atau menghapus

df.columns

```
Index(['No', 'Longitude', 'Lattitude', 'N', 'P', 'K', 'Ca', 'Mg', 'Fe', 'Mn',
       'Cu', 'Zn', 'B', 'b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5',
       'b4', 'b3', 'b2', 'b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
       'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh'],
      dtype='object')
```

Perintah di atas untuk menampilkan daftar kolom pada dataset

```
df['Mg'] = pd.to_numeric(df['Mg'], errors='coerce')
df=df.dropna()
```

df = df.dropna() digunakan untuk menghapus semua baris yang berisi nilai kosong (NaN) dari dataset

## ▼ Feature Selection

```
#X= df[['b2','b3', 'b4', 'b8', 'b11', 'Sigma_VV', 'Sigma_VH']]
#y= df['N']
```

Perintah di atas sengaja di ubah menjadi komentar karena ingin memakai variable lebih banyak

```
# feature selection
X = df[['P', 'K', 'Ca', 'Mg', 'Fe', 'Mn', 'Cu', 'Zn', 'B',
        'b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5',
        'b4', 'b3', 'b2', 'b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
        'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh']]
```

```
y = df['N']
```

Perintah diatas untuk membagi fitur dan juga target

## ✓ Splitting Data

```
#split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Mensplitting data atau membagi data menjadi 20% data testing dan 80% training

```
model = LinearRegression()
model.fit(X_train, y_train)

▼ LinearRegression ⓘ ⓘ
LinearRegression()
```

Perintah Diatas untuk membuat model yang ingin digunakan yaitu linear regression

## ✓ Machine

```
# testing
y_pred = model.predict(X_test)
# evaluasi model
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print("R² squared:", r2)
print("RMSE:", rmse)

R² squared: 0.7549955124484078
RMSE: 0.19392193397227142
```

Perintah diatas untuk mengetahui sebaik apa model regresi memprediksi

```
coeff = pd.DataFrame({
    'Fitur': X_train.columns,
    'Koefisien': model.coef_
})
print(coeff)
```

	Fitur	Koefisien
0	P	8.570837
1	K	0.056014
2	Ca	-0.058380
3	Mg	-0.161095
4	Fe	0.000284
5	Mn	0.000177
6	Cu	0.023202
7	Zn	0.002968
8	B	-0.002092
9	b12	0.327312
10	b11	-0.619437
11	b9	-0.047586
12	b8a	0.064453
13	b8	-0.072659
14	b7	0.981186
15	b6	-0.864947
16	b5	-0.329397
17	b4	1.420614
18	b3	-0.449043
19	b2	-0.357454
20	b1	-0.033418
21	Sigma_VV	0.156595
22	Sigma_VH	-1.466200
23	plia	0.021315
24	lia	-0.021906
25	iafe	-0.078122
26	gamma0_vv	0.457327

```

27 gamma0_vh  2.946416
28 beta0_vv  -0.569281
29 beta0_vh  -0.677599

```

Perintah diatas untuk mengetahui tabel fitur

```

import statsmodels.api as sm
X_sm = sm.add_constant(X)

# tambahkan intercept
model_ols = sm.OLS(y, X_sm).fit()
print(model_ols.summary())

```

OLS Regression Results							
Dep. Variable:	N	R-squared:	0.779	Model:	OLS	Adj. R-squared:	0.768
Method:	Least Squares	F-statistic:	66.10	Date:	Fri, 07 Nov 2025	Prob (F-statistic):	6.69e-163
Time:	15:41:21	Log-Likelihood:	156.68	No. Observations:	592	AIC:	-251.4
Df Residuals:	561	BIC:	-115.5	Df Model:	30		
Covariance Type:	nonrobust						
	coef	std err	t	P> t	[0.025	0.975]	
const	3.8681	0.864	4.479	0.000	2.172	5.564	
P	8.5760	0.465	18.428	0.000	7.662	9.490	
K	0.0273	0.051	0.530	0.596	-0.074	0.128	
Ca	-0.0416	0.023	-1.808	0.071	-0.087	0.004	
Mg	-0.2166	0.063	-3.464	0.001	-0.339	-0.094	
Fe	0.0002	0.000	1.102	0.271	-0.000	0.001	
Mn	0.0002	4.59e-05	4.083	0.000	9.72e-05	0.000	
Cu	0.0229	0.006	4.046	0.000	0.012	0.034	